

ATARI
ST
COMPUTER
Die Fachzeitschrift für den ATARI-ST Anwender.
März '88
DM 7,-
Ös. 56,- Sfr. 7,-
3
ALS DIE BILDER
LAUFEN LERNTEN
-IMAGIC Grafik Compiler
WO ADIMENS
AUFHÖRT
ADITALK ST
OBJEKTORIENTIERTES
ZEICHNEN
mit CAD projekt
STEUERN MIT 24 BIT
I/O-Port im Selbstbau
Video 8 AF
Auto Focus

ST-KOMPLETT



Schneider / Steinmeier

Anwendungen in GFA-BASIC

Heim-Verlag

Verdeutlicht an mehr als 50 Programmbeispielen die Programmierung in **GFA-BASIC**. Alle Programme sind ausführlich dokumentiert, übersichtlich programmiert und damit leicht verständlich. Viele Tips, Tricks und Anregungen können in eigene Programme übernommen werden. Praxisnahe Auswahl der Programmthemen aus vielen Bereichen für jedermann von Interesse.
Aus dem Inhalt: Tips und Tricks zur GFA-BASIC-Programmierung (Verwendung von GEM-Funktionen, Eingaberoutinen, Spriteprogrammierung) · Utility- und Hilfsprogramme (u.a. Kopierprogramme, Mauszeiger-Editor, Sprite- und Füllmustereditor) · Grafik-Programmierung in GFA-BASIC (u. a. 3D-Grafik, Turtlegrafik) · Anwendungsprogramme (u. a. Dateiverwaltung, Vokabeltrainer) · Mathematische Anwendungen (u. a. Statistik, Ableitungen) · Spiele (Alamo, Space-Race, Hamurabi, Klicker)

B-410 Buch DM 49,– D-430 Diskette DM 39,–

ATARI ST

H. Haase

Das große VIP-Buch

Das Kompendium für den Anfänger und den Profi

Heim-Verlag

Wenn Sie das Software-Paket **VIP-Professional** kaufen wollen oder schon besitzen, dann weiht Sie dieses Buch schnell und umfassend in die Geheimnisse dieses Profiprogrammes ein.
VIP-Professional besteht aus den drei Funktionsbereichen
- DATENBANK
- KALKULATION
- GRAFIK

mit denen wichtige und vielfältige Aufgaben hervorragend gelöst werden.
Das Buch enthält komplette Musterlösungen für die Gewinn- und Verlustrechnung und Fakturierung. Anhand dieser Beispiele wird gezeigt, wie Sie alles herausholen, was in VIP-Professional steckt.
Mit diesem Buch können Sie **VIP-Professional** richtig einsetzen und seine Möglichkeiten voll ausschöpfen.

B-408 Buch DM 49,– D-428 Diskette DM 39,–

Mit über 130 Programmbeispielen und Erläuterungen lernen und trainieren Sie das Programmieren in BASIC. Grundlegende Beisp. bis zur ausgereiften Anwendung machen den perfekten Einstieg in die Programmiersprache des GfA-BASIC's leicht. Alle Beispiele werden so dargestellt, daß das Verständnis für die Programmstruktur gefördert wird. Die Progr. sind gut dokumentiert, wobei die hervorragenden Mögl. des GfA-BASIC zur strukturierten Programmierung genutzt werden.
Aus dem Inhalt: Eigenschaften des GfA-BASIC-Dateiverwalt. u. Ordner in GfA-BASIC · Schleifentechnik · Felder · Unterprogramme · Menü-Steuerung · Window-Technik · Zufallszahlen · Seq.- u. RANDOM-Dateien · Textverarbeitung · Sortierprogramme · Fakturiersysteme · Grafik · Spiel uvm. 320 Seiten mit 131 Programmen in GfA-BASIC.

B-407 Buch DM 49,– D-428 Diskette DM 39,–

ATARI ST

Prof. Dr. B. Bollow/K. Reimann

GfA-BASIC Programm-Sammlung

131 Programme in GfA-BASIC

Heim-Verlag

Ein Standardwerk für den ATARI-ST – ideal für Schüler und Studenten! Anhand von zahlreichen Beispielen lernen Sie die Lösung von mathematischen Problemen mit dem ATARI ST kennen. Die zahlreichen BASIC-Programme sind praxisnah ausgewählt worden und können in Schule, Studium und Beruf eingesetzt werden.
Einige der Themen:
- Integral- und Differentialrechnung
- Kurvendiskussion
- Statistik und Wahrscheinlichkeitsrechnung
- Matrizenrechnung
- Radioaktivität
- Relativitätstheorie

u.v.a.m.

B-409 Buch DM 49,– D-429 Diskette DM 39,–



O. Steinmeier

Mathematik und Naturwissenschaft

Heim-Verlag



J. Wilhelm

GEM-Programmierung

Ein Grundwerk für alle Atari-GEM-Programmierer

Heim-Verlag

Ein Grundwerk.
Wer sich mit der Programmierung der GEM-Funktionen vertraut machen will braucht dieses Buch! Es beginnt mit einer Erläuterung des GEM-Aufbaus und führt anhand von Programmbeispielen zum **leichten Verständnis aller auf dem Atari verfügbaren GEM-Funktionen.** Die übersichtliche Gliederung ermöglicht auch ein schnelles Nachschlagen der Funktionen.
Einige der Themen:
– Was ist **GEM** · Die GEM-Bestandteile **VDI** und **AES** · Die **GEM-Implementation** auf dem Atari ST · Aufruf der GEM-Funktionen aus **BASIC, C** und **ASSEMBLER** · Sonstige **Programmiersprachen** und GEM · Die Programmierung der **VDI**-Funktionen · Die **AES**-Bibliothek und die Programmierung ihrer Funktionen · Aufbau eines **Objektbaumes** · Was ist eine **Resource**-Datei? · Viele erläuterte **Beispielprogramme** in BASIC, C und Assembler.

B-404 Buch DM 49,– D-424 Diskette DM 39,–



K. Schneider/O. Steinmeier

Grundlehrgang

Der richtige Einstieg

Heim-Verlag

Das Buch für den richtigen Einstieg mit dem ATARI ST. Auf rund 330 Seiten wird der Leser leicht verständlich in die Bedienung des Rechners eingeführt.
Einige der Themen:

Die Hardware des ATARI ST:
- Aufstellung und Wartung des Computers
- Überblick über die Systemkomponenten
- Das „Innenleben" des Computers

Die Software des ATARI ST:
- Bedienung des GEM-Desktop
- Arbeiten mit Maus, Fenstern und Icons
- Einführung in die Sprachen LOGO und BASIC
- Programmsammlung mit vielen interessanten Beispielen

Diese zweite Auflage des Grundlehrgangs berücksichtigt alle neuen Rechnermodelle der ATARI ST-Familie.

B-400 Buch DM 49,– D-420 Diskette DM 39,–

Zum neuen Basic Interpreter, ein Buch, das mit gezielten Beispielen verständlich den Einstieg in das Basic der Superlative ermöglicht. Ein Muß für jeden Besitzer dieses Interpreters.
Einige der Themen:
- Der Umgang mit dem Editor
- Ausführliche, mit Beisp. versehene Befehlsübersicht
- Die fantastischen Grafikmöglichkeiten (Windows, Sprites, Alertbox, Pull-down Menüs)
- Strukturierte Programmierung – auch in Basic möglich!
- Dateiverwaltung unter GfA-Basic
- GEM Handhabung in Basic
- Vergleich mit anderen Basic-Interpretern – Hilft bei der Kaufentscheidung
- Zahlreiche Übungs- und Anwenderbeispiele
- Mathematik und hohe Genauigkeit

B-405 Buch DM 49,– D-425 Diskette DM 39,–



Bärtels/Egel/Merino/Schneider

Das GfA-Basic Buch

über 550 Seiten

Heim-Verlag

C ist die zweite „Muttersprache" des Atari ST: schnell, komfortabel, kompakt im Code.
„C auf dem Atari ST" ist für Anwender geeignet, die Erfahrungen mit anderen Programmiersprachen gemacht haben. Das Buch behandelt den vollen Sprachumfang von Standard-C und verweist auf BASIC und Pascal.
Besonderer Wert wird auf die Anschaulichkeit und Genauigkeit der Darstellung gelegt. Alle Programmbeispiele sind getestet und **direkt** in den Text übernommen.
Aus dem Inhalt: C-Compiler für den Atari: Digital Research, Lattice, Megamax der Editor · Bedienung des Compilers · Grundlegende Elemente eines C-Programms · Variablentypen · Felder und Vektoren · Ausdrücke · Zeiger · Speicherklassen · Bitfelder · Varianten · Aufzählungen · Dateien · Diskettenhandling · Einbindung von Assemblerprogrammen · Bildschirmgrafik in C · Fehler in den C-Compilern · Tools u.a.

B-406 Buch DM 49,– D-426 Diskette DM 39,–



Michael Sperber

C auf dem Atari ST

Heim-Verlag

BESTELL-COUPON

an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Ich bestelle ____________________
zzgl. DM 5,– Versandkosten (unabhängig von der bestellten Stückzahl)
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ____________________

Straße, Hausnummer ____________________

PLZ, Ort ____________________
Benutzen Sie auch die im ST COMPUTER vorhandene Bestellkarte.

alle Preise sind unverbindlich empfohlene Verkaufspreise



Heim Verlag
Heidelberger Landstr. 194
6100 Darmstadt-Eberstadt
Telefon 06151-56057

Gefahr im Verzug

Sinn eines jeden Editorials ist, ein Thema aufzugreifen, das momentan in aller Munde ist. Trotzdem möchte ich mich einmal mehr mit dem neuen alten Thema Raubkopieren beschäftigen. Der Grund ist eine Nachricht, die mich vor einigen Tagen nachdenklich gestimmt hat.

Amerikas Software-Häuser resignieren

Begonnen hat die ganze Raubkopiererei vor einigen Jahren, als der kleine Bruder des ATARI ST, der ATARI 400/800, auf dem amerikanischen und deutschen Markt erschien. Kurz darauf folgte der Commodore 64, der das Raubkopieren zum Volkssport avancieren ließ. Alle Welt kaufte sich einen Home-Computer zu einem relativ geringen Preis und war dann kaum bereit, Geld für Software auszugeben. Die Folge ist, daß die Verfügbarkeit von möglichen Raubkopien ein Kaufgrund geworden ist. Nichtzuletzt deswegen ist der C-64 zum meistgekauften Home-Computer überhaupt geworden.
Schon zu der Hoch-Zeit des C-64 wurden die zu hohen Preise der Software beklagt, und als die Preise niedriger und niedriger wurden, änderte selbst diese Tatsache die Einstellung der meisten Raubkopierer nicht. Der einzige Grund, warum heute noch Software-Häuser für den C-64 Programme entwickeln, ist die Tatsache, daß selbst bei über einer Million verkauften Rechnern noch ein paar potentielle Käufer vorhanden sind. Man kann sagen, daß die Raubkopierei mit der Entstehung des C-64 gewachsen ist und er vielleicht auch gerade deshalb diesen großen Erfolg gehabt hat.



Wie ergeht es aber einem Rechner, der in einer Zeit des Wilden Westens - sprich der Raubkopiererei à la carte - geboren wird ? Stellen Sie sich vor, sie hätten ein Software-Haus. Nehmen wir weiter an, es gäbe circa 400000 Exemplare des Rechners auf dem Markt, für den Sie Software erstellen. Angenommen, Sie merkten, daß die Raubkopierrate mindestens genauso hoch ist wie auf dem C-64! Was wäre dann ihre Schlußfolgerung ? RICHTIG! SIE WÜRDEN SICHERLICH DAS ENTWICKELN DER SOFTWARE AUF DIESEM RECHNER ANDEREN FIRMEN ÜBERLASSEN, DENN SIE WOLLEN VON IHREM GESCHÄFT LEBEN. Sehen Sie! Und genau das haben viele der Softwarefirmen in den USA, die für den ATARI ST Software produzieren, getan. Bei einem Besuch der Messen in den Vereinigten Staaten erfuhren wir unter anderem von den Firmen HABA und ACTIVISION, daß sie sich ganz vom ST-Markt zurückziehen werden und das, obwohl HABA gerade mehrere große Programme für den ST fertiggestellt hat. HABA wird diese Software mit großer Wahrscheinlichkeit einem Interessenten verkaufen und dem ATARI ST den Rücken zukehren. Die Begründung liegt in dem in den USA stark 'Spielegedrängten' und damit professioneller Software nicht aufgeschlossenen ATARI-ST-Markt. Zusammengefaßt: Zuwenig Interesse an professioneller Software - IBM läßt sich nicht so einfach in den USA aus den Köpfen der Anwender verdrängen - und der restliche (Spiele-)Markt, an dem beispielsweise ACTIVISION interessiert ist, lohnt sich aufgrund der hohen Raubkopierrate nicht.

Europa gibt nicht auf!

Und wie sieht's hier in Deutschland aus? Glücklicherweise haben die europäischen Anwender erkannt, daß auf dem ST auch professionelle Software zu haben ist. Daher bricht der Markt des ST in Europa (noch) nicht zusammen. Außerdem gilt der ST in Europa nicht so sehr als Spielerechner wie in den Staaten. Der ST ist in den USA der C-64-Nachfolger (Spiele, Spiele, Spiele ...), und der AMIGA sucht sich ein paar Anteile am professionellen Markt. Seltsamerweise ist genau das Gegenteil in Europa festzustellen. Man kann für den AMIGA hoffen, daß die Raubkopierer Mitleid mit ihrem Rechner haben, sonst ist der Markt in Europa auf dem AMIGA sehr schnell genauso kaputt wie in den USA auf dem ATARI ST.
In der Hoffnung, daß die Raubkopierer endlich ein Einsehen bekommen und merken, daß sie sich tatsächlich selbst schaden, programmiere ich - nachdenklich - weiter an meinem ATARI ST.

Stefan Höhn

I N H

Allgemeines

Software

Hardware

Grundlagen

Anwendungen

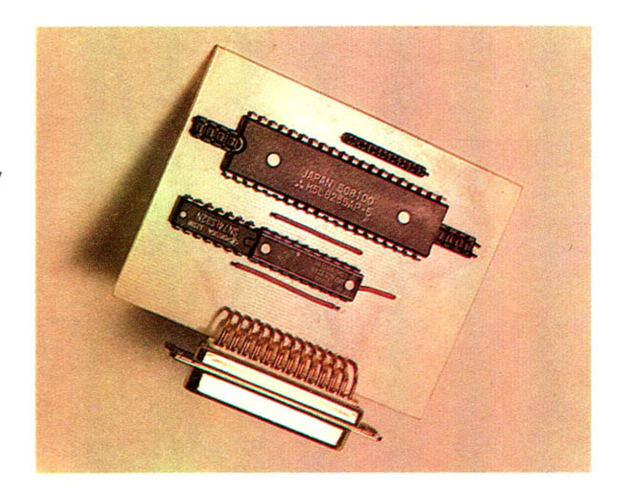
Programmierpraxis

Aktuelles

A L T

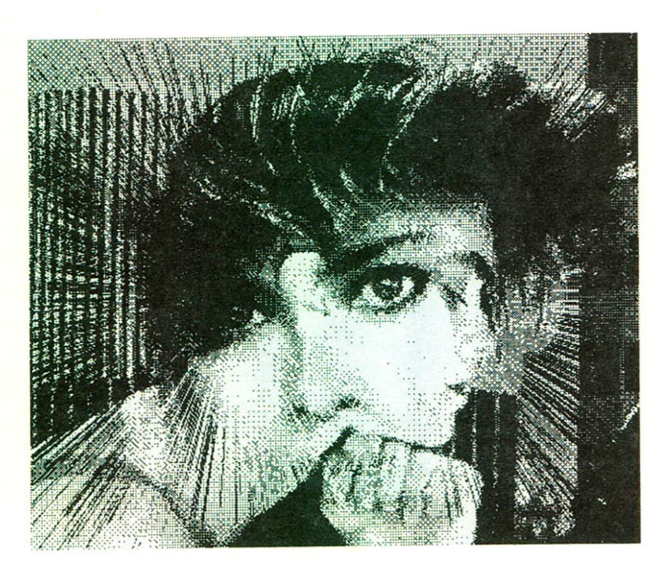
24-Bit -Interface am Druckerport

Eine der Kinderkrankheiten des ST ist das Fehlen von I/O-Slots zur Ansteuerung externer Geräte wie z.B. Meßinstrumente oder Roboter. Mit einer kleinen Schaltung, die auch für den kleinen Geldbeutel verkraftbar ist, und ein wenig Geschick haben auch Sie die Möglichkeit Ihre Kaffemaschine zum Frühstück zu programmieren



Wo ADIMENS aufhört, macht ADITALK weiter!

Wir stellen Ihnen die Datenbankkommandosprache ADITALK vor, die sowohl eigenständig als auch als Ergänzung zu ADIMENS ST zu benutzen ist. Wer sich näher mit DBase-ähnlichen Programmen beschäftigt hat, wird sich auf Anhieb bei ADITALK wohlfühlen. Eigene Programmentwicklung ist mit einem Compiler problemlos möglich.

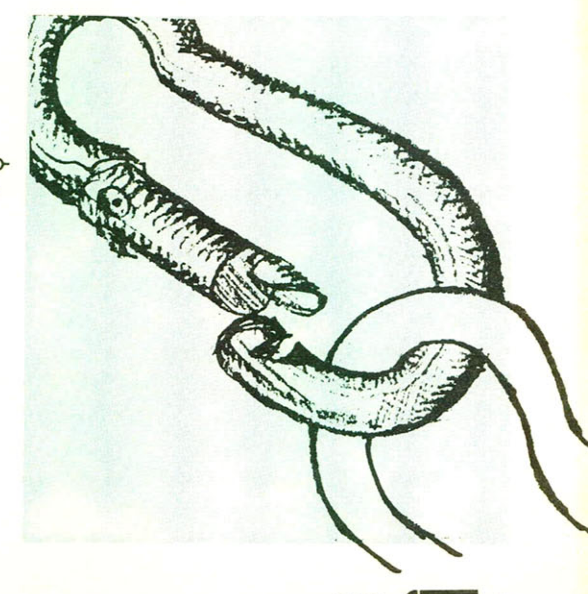


IMAGIC Grafik Compiler - nach langem Warten, endlich da!

Angekündigt und vorgesehen war der Testbericht für die Juli/August-Ausgabe letzten Jahres, dann mußten Teile des Grafik Compilers neu geschrieben werden. Doch jetzt ist er endlich fertig. Lesen Sie alles über die endgültige Version dieses grafischen Entwicklungspaketes, dessen Produkte bestimmt schon einigen auf der ein oder anderen Messe aufgefallen sind.

Neuer Linker für das GST-Format

Für alle, denen der bisherige Linker für das GST-Format, den ja doch einige Programmiersprachen benutzen, zu langsam und nicht ausreichend war, listen wir einen neuen, schnellen Linker ab. Er ist mit dem Lattice C-Compiler erstellt worden. Erklärt werden Begriffe wie Interpreter, Compiler, Linker, das GST-Format und die Arbeitsweise eines Linkers.



NEWS

Probleme mit dem Diskettenformat?

Für alle Umsteiger von 8-Bit-CP/M-Computern, einschließlich Schneider JOYCE und CPC, auf die 16-Bit-Computer mit MS DOS-Betriebssystem und auf den ATARI ST bietet das Mathematische Softwarebüro Bernd Drost eine einfache und preisgünstige Lösung:
Zum Preis von DM 28,- inkl. Diskette erhält der Besteller nach der Einsendung seiner Originaldiskette eine Diskette im MS-DOS- bzw. ATARI-Format. Staffelpreise bei mehreren Disketten sind möglich. Als zusätzliche Option können Wordstar-Dateien in ASCII-Dateien und die deutschen Umlaute des ASCII-Zeichensatzes in die Umlaute des IBM- bzw. ATARI-Zeichensatzes umgewandelt werden. Die CP/M-Formate umfassen alle Laufwerksgrößen von 3" bis 8".

Mathematisches Softwarebüro
Bernd Drost
Schulstr. 67
6382 Friedrichsdorf

Video und Computer

Die Heidelberger Firma FTS hat einen ST-Videorecorder-Controler für Geräte vom Typ U-Matic und VHS der Firmen SONY, Panasonic und JVC angekündigt. Es steht eine einfach zu handhabende, flexible Steuersprache zur Verfügung, die aus jeder Programmiersprache heraus benutzbar und vollständig dokumentiert ist.
Zusätzlich wird noch einiges an Software rund um den Videobereich angeboten, so z.B. VIDEO-INFO, ein voll integriertes Softwarepaket mit Cassettenverwaltung, computergesteuertem Schnitt usw.

FTS Computer- & Video-Vertriebsgesellschaft mbH
Steinbachweg 1
6900 Heidelberg 1
Tel.: 06221/802420

Neuer Editor von Pahlen & Krauss

PKSEDIT ist ein neuer GEM-unterstützter, schneller Programmeditor, der das Edieren von max. vier Dateien in vier Fenstern gleichzeitig unterstützt. Die max. Zeilenlänge beträgt 4096 Zeichen, die max. Größe einer Datei ist nur durch den Hauptspeicher begrenzt. Es lassen sich ganze Dateibäume nach Ausdrücken durchsuchen. Das Erstellen von Kreuzverweislisten ist ebenfalls möglich.
Das Programm arbeitet mit der Megamax-Shell zusammen und kann beliebige von Compilern und Assemblern erzeugte Fehler-Dateien als Suchvorschrift in Dateien verwenden und automatisch anspringen.
Alle Features aufzuzählen, würde hier den Rahmen sprengen. Weitere Informationen bei:

Pahlen & Krauss Software
Kolonnenstr. 29
1000 Berlin 62

NEUES VON COMPUTERWARE

Aus Köln erreichte uns die Nachricht, daß für die MT-C-Shell von Computerware jetzt ein Virtual Screen Handler (VSH)-Manager lieferbar ist. Damit kann man in mehreren Fenstern gleichzeitig mit der MT-C-Shell arbeiten. Durch ein Accessory hat man jederzeit Zugriff auf die Leistungen der Shell.
Für 1988 wird Regent Word II, eine funktionelle Textverarbeitung mit großer Flexibiltät, angeboten. Ein integrierter Taschenrechner und ein Wort-Prüf-Programm gehören ebenso dazu wie die Möglichkeit, Serienbriefe zu schreiben.
Ebenfalls 1988 erhältlich ist die Datenbank The Informer. Sie ist GEM-eingebunden und kann Texte und Grafiken verwalten. Alle Funktionen sind über Maus oder Tastatur leicht erreichbar. Die Daten können als GEM-Formular oder als Tabelle dargestellt werden.

Computerware
Gerd Sender
Moselstr. 39
5000 Köln 50
Te.: 0221/392583

Signum!-Manager

Mit dem Hilfsprogramm Signum!-Manager werden die verschiedenen zu Signum! (1 und 2) gehörenden Programme durch eine Shell verwaltet. Zusätzlich kann man einen Texteditor für ASCII-Texte, die Programme STAD, Doodle und drei zusätzliche Fremdprogramme aufrufen. Bei Anforderung ist lediglich ein Unkostenbeitrag von DM 30,- zu erstatten.

Dietmar Rabich
Dövelingsweg 2
4408 Dülmen

Assembler-Tutor

Der Hüthig Verlag bietet einen Assembler-Programmierkurs zum Preis von DM 38,- auf Diskette an. Der Kurs umfaßt 29 Lektionen und mehrere Bildschirmtafeln, auf denen u.a. die erlaubten Adressierungen und die Tastaturcodes abrufbar sind.

Mit im Lieferumfang ist ein PD-Assembler und Debugger, um das Gelernte gleich in die Praxis umzusetzen. Ferner ist eine resetfeste RAM-Disk auf der Diskette enthalten. Benötigt wird mindestens ein 512 KB-Rechner mit ROM-TOS. Der Kurs läuft in Farbe und Schwarzweiß.

Assembler Tutor für alle
ATARI ST Computer
von Heinrich Kersten
Hüthig Verlag
ISBN 3-7785-1508-X

Neues von GFA Systemtechnik

Ein Farb- und ein Monochromkonverter werden neuerdings von GFA Systemtechnik vertrieben. Dabei handelt es sich um rein softwaremäßige Umwandlungen der jeweiligen Grafikauflösungsstufen (niedrige, mittlere und hohe Auflösung).
Man kann mit dem Farbkonverter auf jedem monochromen ATARI-Monitor (SM124) auch die mittlere und die niedrige Auflösung (natürlich in Schwarzweiß) benutzen. Die Farben werden dabei in Graustufen umgerechnet. Das genaue Gegenstück bildet der Monochromkonverter, der es erlaubt, auf einem Farbmonitor (SC1224) die hohe Auflösung zu benutzen. Der Preis soll pro Konverter bei DM 59,- liegen.

GFA Systemtechnik GmbH
Heerdter Sandberg 30
4000 Düsseldorf 11
Tel.: 0211/588011

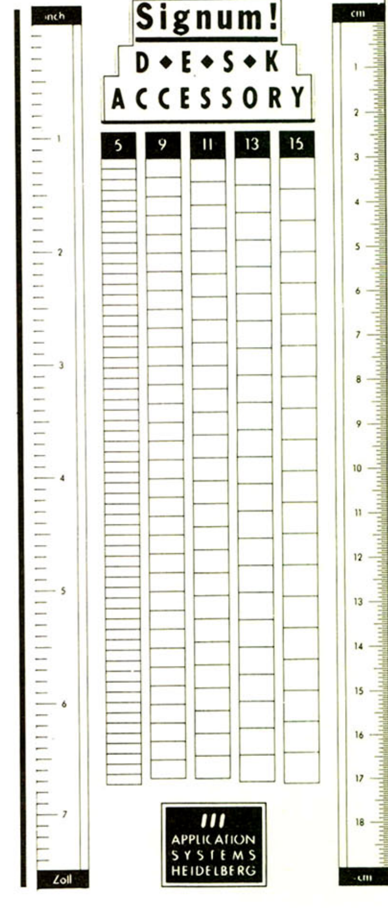
OMIKRON. Compiler 2.0

Accessories waren bislang in BASIC ein Ding der Unmöglichkeit. Dem jetzt erhältlichen Update des OMIKRON.-Compilers macht das Erzeugen von Accessories keinerlei Probleme mehr.
Die weitere Neuheit ist die kürzbare Library mit dem passenden Namen CUTLIB. Bislang wurde die komplette BASLIB entweder angebunden oder nach Programmstart nachgeladen. Die CUTLIB benötigt eine minimale Länge von 4 KByte und erweitert sich entsprechend mit den benötigten Routinen. Alle bis zum Erscheinungszeitpunkt registrierten User erhalten das Update kostenlos zugesandt. Nichtregistrierte User erhalten das Update zum Unkostenbeitrag von 10.- DM.

Statistisch gesehen...

In vielen wissenschaftlichen Anwendungen sind statistische Auswertungen vonnöten. OMIKRON.-Software bietet daher eine umfangreiche Bibliothek für ihren Basic-Interpreter/Compiler an. Darin enthalten sind neben Befehlen der Grundstatistik, umfangreiche Verteilungs- und Prüffunktionen, Zufallsgeneratoren, Konfidenzintervalle, Tests auf Übereinstimmung mit Sollwerten, Vergleich von Stichproben, Anpassungstests, Mehrfeldertafeln, Varianzanalyse, Regressionsprobleme und Zeitreihenanalysen. Zur grafischen Darstellung der errechneten Werte stehen Befehle für Kreis- und Blockdiagramme, sowie diverse Befehle zum Plotten von Kurven zur Verfügung. Die Statistik-Library ist zum Preis von DM 79.- zu erhalten bei:

OMIKRON.-Software
Erlachstraße 15
D-7534 Birkenfeld 2
Tel.: 07082/5386



Signum! Desk Accessory

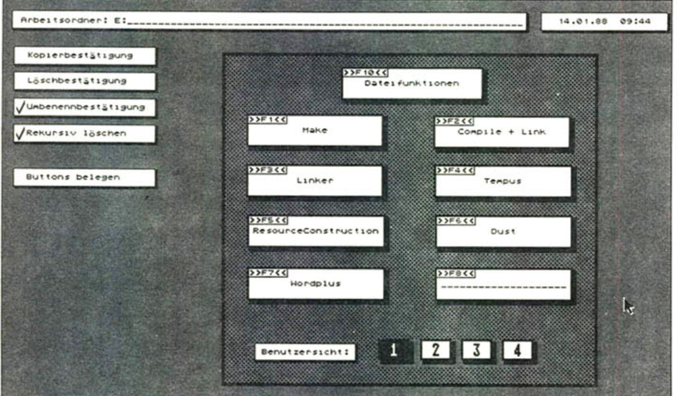
Hinter diesem Namen verbirgt sich nicht eines der ersten Accessories, das sich endlich nach langem Widerstreben aus Signum! in gewohnter GEM-Manier aufrufen läßt, sondern es handelt sich dabei um ein Skalenlineal, das das genaue Ausmessen einer Vorlage für ein Layout erlaubt. Das Lineal ist für DM 20,- zzgl. Versandkosten zu beziehen.

Application
Systems /// Heidelberg
Englerstr. 3
6900 Heidelberg
Tel.: 06221/300002

NEUE HARDWARE

bietet die Firma TJ-Soft und Hardware an. Im Programm hat sie u.a. einen 8-Bit AD-DA-Wandler, der an den Centronics-Port angeschlossen wird. Der AD-Wandler hat eine Wandlungsrate von extern 10000 und intern 100 Wandlungen/Sekunde und einer Auflösung von 10 mV/Bit (Meßbereich 0 bis 2,55), der DA-Wandler eine Ausgangs-Gleichspannung von ebenfalls 10 mV/Bit. Geliefert wird die Karte für DM 189,- zusammen mit diversen GFA BASIC- Hilfsprogrammen.
Eine weitere AD-DA-Wandlerkarte wird ab Februar für DM 248,- erhältlich sein. Hier erfolgt der Anschluß über den Modul-Port des ST. Sie hat 8 analoge programmierbare Eingänge und zusätzlich 32 frei programmierbare I/O-Ports mit eigener Spannungsversorgung. Auch hier wird diverse Software in GFA BASIC mitgeliefert.
Ebenfalls wird von der gleichen Firma ein 8-Bit Schaltinterface für DM 128,- angeboten. Der Anschluß erfolgt 8 Bit parallel über den Centronics- oder den Joystick-Port des ST. Die Ausgänge sind potentialfrei und über Relais vom Netz getrennt. Die Schaltleistung pro Kanal beträgt 500 Watt.

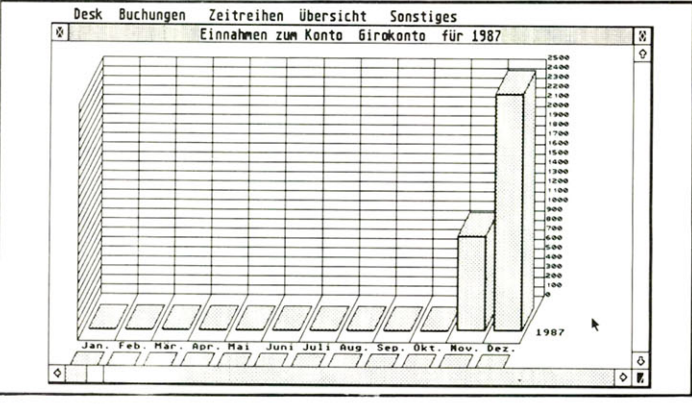
TJ-Soft und Hardware
Postfach 10 05 20
3250 Hameln 1



Neue Finanzbuchhaltung für den ST

Unter dem Namen 'Last Cash' ist ab sofort eine Finanzbuchhaltung für den ST für DM 79,- erhältlich, die sich hauptsächlich an Privatleute wendet, aber auch für Freiberufler und Kleinunternehmer geeignet ist. Sie ist voll GEM-unterstützt, bietet grafische Auswertungsmöglichkeiten und ist universell einsetzbar.

Knöpke Software
Gartenstr. 10
3339 Söllingen
Tel.: 05354/8203



Grafische Übersicht aus "Last Cash"

PC ditto jetzt in Euro Version 3.64

Der MS-DOS-Emulator PC ditto ist jetzt in einer Euro Version 3.64 lieferbar. Diese Version ist an die europäischen TOS-Versionen, also auch an das deutsche TOS, angepaßt und hat auch in punkto Geschwindigkeit einiges mehr zu bieten. Ferner werden mehrere Harddisk-Partitionen (C,D,E usw.) unterstützt. Nähere Informationen unter der untenstehenden Bezugsadresse oder direkt auf der CeBIT ´88 in Hannover (Halle 17 Stand A70).

Merlin Computer GmbH
Industriestr. 26
6236 Eschborn
Tel.: 06196/481811

NEUE PROGRAMMIERSPRACHE MIT HOHER GENAUIGKEIT

Die kaiserslauterner Firma SSE bietet eine ganze Reihe von Produkten an. Da wäre zuerst einmal die Programmiersprache 'Calculus'. Sie bietet eine Pascal-ähnliche Umgebung und ist voll GEM-unterstützt. Ein besonderes Feature ist u.a. die nur durch die Speichergröße begrenzte Rechengenauigkeit, wodurch es vor allem für den naturwissenschaftlichen Bereich sehr interessant werden dürfte. Der Preis beträgt DM 298,-

Ein weiteres Produkt ist der 24 Kanal-MIDI-Sequenzer 'Stencer', der wohl für den Preis von DM 198,- ganz enorme Möglichkeiten bietet und in direkter Konkurrenz zu den bekannten ST-MIDI-Sequenzern stehen könnte. Nebenbei ist auch ein Notendruck möglich.
Ebenfalls von dieser Firma ist ein Laufwerks-Speeder erhältlich, durch den der Datenzugriff enorm beschleunigt wird. Der Speeder wird zusammen mit einer resetfesten RAM-Disk geliefert.

Beide Programme werden über GFA Systemtechnik für DM 59,- vertrieben werden.
Für DM 59,- ist eine C-Shell erhältlich, die sich ganz besonders für den Megamax C eignet.

SSE
System & Software Engineering
Postfach 1531
6750 Kaiserslautern

LESEN SIE SCHEIBENKLEISTER!



SCHEIBENKLEISTER - MASSENSPEICHER AM ST.
Alles über Floppies, Festplatten usw. von Claus Brod und Anton Stepper.
Was steht drin?



Kursteil (für die ganze Familie):
- Floppyprogrammierung mit allen erlaubten und unerlaubten Mitteln (per BIOS, XBIOS, GEMDOS und direkter Controllerprogrammierung);
Kopierschutz, Aufzeichnungsverfahren, Datenstrukturen auf der Diskette
- Hardwaredokumentation zu Floppy und Festplatte (Anschluß von Fremdlaufwerken, Justierung, Reparaturhinweise)
- Festplatte: Prinzip, Controller, Programmierung

Nachschlageteil (für Programmierer):
- Hard- und Softwarereferenz zu DMA-Chip, Floppycontroller, Festplattencontroller
- GEMDOS-, BIOS- und XBIOS-Funktionen zur Massenspeicherprogrammierung (auch als GFA-BASIC-Bibliothek auf Diskette)

Software (für alle, fertig zum Anwenden mit kompletten Anleitungen) :
- TED, der Trackeditor: Formate analysieren, ändern, erstellen
- SED, der RAM-Disk/EPROM-Disk/Floppydisk/Harddisk-Monitor: Ordnernamen ändern, gelöschte Dateien retten, spezieller Harddiskmonitor für direkten Festplattenzugriff (eigene Formatierroutine für zwei MB mehr)
- neue HYPERFORMAT-Version 3.0: MS-DOS -kompatibles Format, bis zu 950 KB auf doppelseitiger Diskette, superfixe Formatierroutinen (optional unter 20 Sekunden für doppelseitige Disketten), Schnelladeformate
- Steprateneinstellung, Konvertierung von Disketten auf Schnelladeformat
- Assemblerroutinen für direkten Floppy- und Festplattenzugriff zum Einbinden in eigene Programme
- lauffähige Programme mit Quelltext auf Diskette

Ca. 600 Seiten, Buch mit Diskette für 59DM, erscheint Ende Februar

---

HIERMIT BESTELLE ICH___EXEMPLARE VON "SCHEIBENKLEISTER. MASSENSPEICHER AM ST".
MIT DISKETTE FÜR DM 59.--
ANRUF GENÜGT: 06196/481811.
MO-FR 9-13 UND 14-17 UHR
SCHRIFTLICHE BESTELLUNG NUR GEGEN

VORKASSE ODER NACHNAHME (VERSANDKOSTEN DM 5.50: BEI NACHNAHME ZUZÜGLICH DM 4.70)
NAME:______
VORNAME:______
STRASSE:______
ORT:______
UNTERSCHR. :______



MERLIN COMPUTER GMBH
INDUSTRIESTRAßE 26
6236 ESCHBORN
TEL. 06196/481811

AT-Tastatur am ST

Einige neue Produkte bietet die Firma 3K EDV-Entwicklungen für den ATARI ST an. AT-A-ST ist eine Adapterbox, die den Anschluß jeder AT-kompatiblen Tastatur an den ST ermöglicht. Die ST-Tastatur bleibt dabei funktionsfähig. Angeboten werden zwei Versionen: Die steckbare Fertigversion mit Netzteil zum Preis von DM 298,- sowie eine Einbau-Version zum Preis von DM 199,-.

Mit P-SWITCH lassen sich bis zu vier Drucker gleichzeitig an die Centronics-Schnittstelle des ST anschließen, die Umschaltung erfolgt softwaremäßig. Die beigefügte Diskette enthält ein Accessory, das neben der Umschaltmöglichkeit auch für jeden Drucker einen eigenen, konfigurierbaren Spooler bereitstellt, so daß alle Drucker gleichzeitig drucken können. Außerdem können die Drucker auch aus einem Text heraus umgeschaltet werden. P-SWITCH kostet in der Ausführung für zwei Drucker DM 188,-, für vier Drucker DM 228,-.

AUTOMON ist eine Umschaltbox für einen Monochrom- und einen Farbmonitor. Weiterhin sind Anschlüsse für Audio Aus- und Eingang vorhanden. Als besonderes Bonbon kann die Umschaltung zwischen Farb- und Monochrommonitor auch per Software erfolgen. Auf der beigefügten Diskette befindet sich als Beispiel dazu ein kommentiertes GFA BASIC-Programm sowie Treibersoftware für CYBERMATE (CAD 3D). Weiterhin wird die Umschaltoption auch von dem DTP-Programm Calamus unterstützt. AUTOMON kostet DM 79,-.

3K EDV-Entwicklungen
Hülser Str. 76
4154 Tönisvorst 1
Tel.: 02151/700522

Ersatz für das GEM Desktop

Mit epsiMenü bietet die Epsilon GmbH ab Frühjahr '88 eine neue Benutzer-Shell für den ST an. Mit ihr sind alle Dateifunktionen des Desktops nachgebildet und einige Funktionen gehen sogar über die Funktionalität des Desktops hinaus. So ist z.B. das Kopieren von einem Ordner in einen anderen durch einfaches Umbenennen wesentlich beschleunigt worden, eine neue Fileselektorbox erlaubt auch die Auswahl mehrerer Dateien, ein Batchprozessor wurde integriert, es können mehrere Dateitypen wie beim PC GEM für ein Programm angemeldet werden und vieles mehr. Der Preis war leider noch nicht zu erfahren.

Epsilon GmbH
Durlacher Allee 53
7500 Karlsruhe 1
Tel.: 0721/616474

Länder dieser Erde

Lernen fällt meist schwer und hat nicht immer den gewünschten Erfolg. Das Programm 'Länder dieser Erde' ist ein Lernprogramm um die vielen, vielen Länder, die es nun mal auf diesem Planeten gibt, kennenzulernen. Dazu werden die einzelnen Erdteile auf dem Bildschirm dargestellt und die Länder per Mausklick abgefragt. Die Arbeitsweise beruht auf dem Prinzip des 'gehirn-gerechten Lernens' von Vera F. Birkenbihl, die auch durch mehrere Bücher bekannt wurde. Der Preis des Produktes liegt bei DM 39.-

LERN PARTNER
Jahnstraße 9/1
7535 Königsbach-Stein
Tel.: 07232-4293



Nicht vergessen! Besuchen Sie die ST-Computer Redaktion auf der CeBIT'88 in Hannover.

ATARI ST und Yamaha FB-01

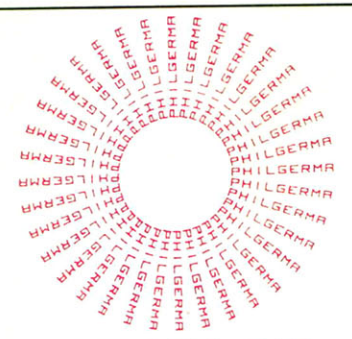
Für den ST ist ein Soundeditor vorgestellt worden, der alle 79 Parameter des FB-01 mit bedienungsfreundlichen 'Schiebereglern' bedienen kann. Man kann seine Frequenzen, Modulationen und Hüllkurven frei wählen. Die erzeugten Klänge können auf Diskette abgespeichert werden. Zusätzlich ist ein Editor für die Konfigurationen enthalten, mit dem sich zahlreiche musikalische Einstellungen für das Keyboard verändern lassen. Auch diese Daten lassen sich speichern. Der Preis beträgt DM 85,-.

Softwarevertrieb
Dipl.-Ing. Peter Vogel
Breslauer Str. 21
2240 Heide
Tel.: 0481/3351

Preissenkung bei GTI

Die Berliner Gesellschaft für technische Informatik (GTI) hat ab 1. Januar den Preis für ihr IEC-Bus-Interface CONTROLLER 488 ST/RS gesenkt. Das intelligente Interface wird in Zukunft mit erweiterter Betriebssoftware ausgestattet, die es erlaubt, das Gerät wahlweise an die MIDI- oder die RS232C-Schnittstelle anzuschließen. Zur Konfiguration des Gerätes müssen lediglich zwei Brücken gesteckt werden. Die Einstellung der Übertragungsraten und des Übertragungsformates erfolgt automatisch. Der neue Verkaufspreis beträgt jetzt DM 1795,50.

GTI mbH
Unter den Eichen 108a
1000 Berlin 45
Tel.: 030/8 31 50 21/22



PHILGERMA IHR SPEZIALIST FÜR ATARI COMPUTERSPRACHEN

ATARI ST

LATTICE C (Metacomco) – Neueste Version 3.04 des bewährten Standard-Compilers der IBM-Welt. Voller Kernigham/Ritchie-Standard. Floating-Point-Arithmetik mit 16 Stellen Genauigkeit. Natürlich die VDI/AES Funktionen, ein sehr guter Resource-Construction-Editor. Utilities (Symbolischer Debugger, MAKE, Shell, Disassemb.) und über 320 UNIX-ähnliche Routinen. Dieses Entwicklungspaket wird mit 600 Seiten starkem deutschen Handbuch geliefert .... DM 298,00

N E U MCC PASCAL2 (Metacomco) – Pascal Compiler ISO 7185 Standard. Schneller 1-Pass Compiler. 64-Bit IEEE Fließpunktarithmetik. Natürlich die VDI/AES Funktionen, ein sehr gutes Resource-Construction-Editor. Utilities (Symbolischer Debugger, MAKE, Shell, Disassemb.) MCC Pascal2 Programme können mit anderen Sprachen gelinkt werden. Deutsches Handbuch 600 Seiten ...... DM 248,00

MCC ASSEMBLER (Metacomco) – Professioneller Makro Assembler, der den vollen Motorola 68000 Instruktion Set unterstützt. Mit Editor, Linker und TOS-Libraries, AES, VDI. Ausführliche Fehlermeldungen. Deutsches Handbuch .... DM 168,00

CAMBRIDGE LISP (Metacomco) – Interpreter und Compiler mit dem Sprachumfang, den man von Großrechnern gewöhnt ist. Volle REAL-Arithmetik 16 MByte Adressraum. Handbuch 330 S. ..... DM 448,00

BCPL Interpreter (Metacomco) – Eine leistungsfähige Sprache, besonders für die Systemsoftwareentwicklung. Mit GEM-Bibliothek .... DM 348,00

APL 68000 Interpreter (MicroAPL) – Eine sehr schnelle Version dieser von IBM-Rechnern bekannten Sprache. Info anfordern .... DM 448,00

MODULA-2 (TDI) – Diese umfangreiche Modula Implementierung vereinigt die Vorteile von Pascal mit neuen Sprachelementen. Integriertes Programmierentwicklungssystem. Compiler, Editor, Debugger und GEM Einbindung. V 3.01 .... DM 248,00

MODULA 2 Developer zusätzlich den Resource-Construction Editor, RAM-Disk, Spooler und umfangreicher Bibliothek V 3.01 .... DM 398,00

MODULA 2 Commercial zusätzlich alle Utilities im Quellkode V 3.01 .... DM 698,00

OMIKRON BASIC (Omikron) – Ein sehr schneller 'Halbcompiler', der die strukturierte Programmierung unterstützt. Rechengenauig. 19 Stellen 99%-ig kompatibel zu M-Basic, kompl. GEM-Library, alle AES/VDI-Funktionen verfügbar, Extras wie Masken-Input, Sort- und Matrizenbefehle, mit 180-seitigem deutschen Handbuch .... DM 178,00
OMIKRON BASIC Compiler .... DM 178,00

PRO PASCAL Compiler .... DM 298,00
PRO FORTRAN77 Compiler .... DM 348,00
AC FORTRAN 77 absoft .... DM 448,00
TRUE BASIC Interpreter .... DM 248,00
TRUE BASIC Runtime Package ..... DM 248,00
TRUE BASIC 3D-Grafik Packet ...... DM 128,00
AC-FORTRAN .... DM 398,00

NATÜRLICH HABEN WIR AUCH SPIELE:
Terrorpods DM 69,–; Barbarian DM 69,–; Spy vs. Spy DM 79,–; Star Wars DM 59,–; Tracker DM 69,–; **Test Drive** DM 99,–; Super Sprint DM 59,–; Goldrunner DM 59,–; The Guild of Thieves DM 59,–; Dungeon Master DM 79,–; Psion CHESS DM 69,–; Arkanoid DM 39,–; The Bard's Tale I DM 99,–; Rampage DM 39,–; Frost Byte DM 59,–; Pawn DM 69,–; Defender of the Crown DM 89,–; Phoenix DM 49,–; Jupiter Probe DM 39,–; Star Trek DM 49,–; Western Games DM 59,–; Wizball DM 49,–; **Jinxter** DM 69,–; Marble Madness DM 89,–; **Flight II** sw & farbe DM 99,–; **Asterix im Morgenland** DM 69,–
– Alle Infocom Text-Adventures vorrätig –

ST PASCAL plus Compiler .... DM 248,00
MARK WILLIAMS C Compiler .... DM 348,00
MEGAMAX C Compiler .... DM 448,00
GFA BASIC Interpreter .... DM 98,00
GFA BASIC Compiler .... DM 98,00
GFA BASIC 68881 Interpreter/Comp. .. DM 348,00
LDW BASIC Compiler Atari Basic komp. DM 98,00

SALIX PROLOG Interpreter .... DM 198,00
FORTH ST+ von Data Becker .... DM 298,00
PROFIMAT ST Assembler Data Becker DM 98,00

Star Writer ST Textverarbeitung ...... DM 198,00
WordPerfext Textverarbeitung .... DM 790,00
1st WORD PLUS deutsch .... DM 198,00
1st Proportional Utility .... DM 88,00
SIGNUM 2 Text- und Grafik-Programm . DM 428,00
PUBLISHING PARTNER .... DM 498,00

ADIMENS Datenbank deutsch neu .... DM 198,00
TIM Buchführungsprogramm V 1.1 .... DM 298,00
LOGISTIX Tabellenk., Datenb., Grafik .. DM 378,00
K-SPREAD 2 Tabellenkalkulation ...... DM 228,00
K-GRAPH 2 Grafik + Statistik .... DM 148,00
K-COMM 2 Terminalprogramm VT100 .. DM 148,00
dBMAN Datenbank deutsch .... DM 399,00
T.L.D.U. the last disk utility .... DM 148,00
CAD-3D bewegte 3D-Grafik .... DM 298,00
CADproject CAD-Programm .... DM 298,00
CADproject Vollversion m. Plottertreiber u. vollautom. Bemaßung .... DM 798,00

AUSZUG AUS UNSERER HARDWARELISTE:
Einzellaufwerk 3,5" 720KB .... DM 398,00
Doppellaufwerk 3,5" 2★720KB .... DM 698,00
Einzellaufwerk 5,25" 40/80 Spuren .... DM 498,00
Speichererweiterung auf 2,5 MByte .... DM 998,00
10 Disketten 3,5" 2DD no name ...... DM 29,00

HARDDISK XEBEC Alle Geräte im Metallgehäuse m. Platz für 2 Laufwerke, 8 Partitions möglich
Harddisk 40 MB 28mS komplett ..... DM 2598,00
Harddisk 70 MB 28mS komplett ..... DM 3998,00
Harddisk 40 MB mit Tape-Streamer ... DM 4398,00
K-MAX Transputerboard .... DM 2598,00
Cascade OCCAM Transputer-Software DM 598,00
OMEGA Farbgrafikkarte 1024★512 ... DM 2698,00

PHILGERMA PRÄSENTIERT

BASiCALC

- einfache Bedienung unter GEM; Online-Hilfe aufrufbar.
- Programm und Handbuch komplett in deutsch.
- über 130 mathematische und finanztechnische Funktionen ähnlich Lotus 1-2-3
- bis zu 10 000 Zeilen oder 676 Spalten nach vorhandenem Speicherplatz.
- Datenaustausch mit anderen Programmen über ASCII- oder DIF-Dateien.
- läuft auf jedem ATARI ST in Farbe oder s/w; einfache Druckeranpassung.

Die GEM Produktfamilie jetzt auch auf dem ATARI ST verfügbar!

- **GEM** Diary (incl. GEM System u. GEM Desktop) 169,– DM
- **GEM** Write + **GEM** Paint .... 399,– DM
- **GEM** Graph .... 369,– DM
- **GEM** Draw Plus .... 369,– DM
- **GEM** Wordchart .... 369,– DM

Dazu wird benötigt:
GEM SYSTEM SOFTWARE V. 2.2
und 1 MB RAM-Speicher
GEM System 2.2 wird geliefert zus. mit GEM Diary.

OS-9

von CUMANA
auf dem ATARI ST implementiert
mitgeliefert: ROM-Modul
Compiler: C.Pascal, Basic, Assembler, Textverarbeitung
Datenbank, Tabellenkalkulation, Echtzeituhr, Handbücher
alles zus. **1598,– DM**

Supercharger MS-DOS-EMULATOR

volle PC-Kompatibilität
durch eigenen 8086-Prozessor
(8 MHz, 1 MB RAM, Sockel f. 8087-Coproz.)
MS-DOS 3.2, IBM-BIOS-Interrupts, serielle Schnittstelle, Clock;
m. vielf. Erweiterungsmöglichk.
für **nur 698,– DM**

PC-DITTO

Software-MS-DOS-Emulator

der bekannte MS-DOS-Emulator aus den USA, eingetroffen in der Version für s/w und Farbe!
Hohe Kompatibilität für nur
198,– DM

CAD PROJECT

Konstruieren mit dem ST

Normalversion .... 298,– DM
Vollversion incl. Plottertreiber u. vollautomat. Bemaßung ... 798,– DM
Demodiskette .... 10,– DM

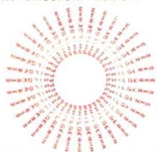
Fordern Sie unsere **Preisliste** an! ·
Händler bitte Händlerliste anfordern.

Tel. Bestellungen von 10°° bis 18³° Uhr
4,80 DM Versandkostenanteil bei Bestellungen unter 200,– DM; Nachnahme 3,20 DM.
Lieferung ins Ausland nur gegen Vorkasse.

Testen Sie alle Produkte in unserem Laden!

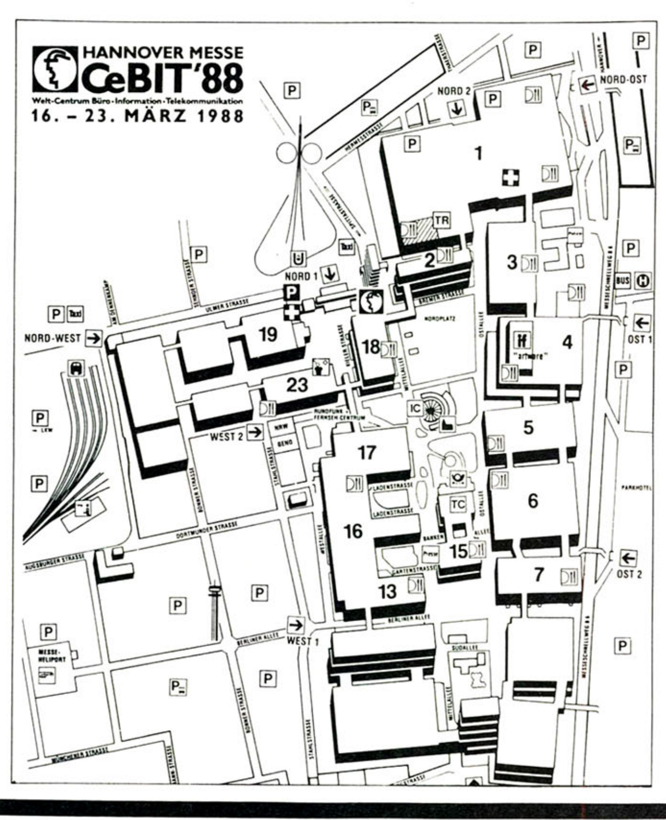
Achtung neue Adresse!

PHILGERMA
GmbH & Co. KG
Barerstr. 32
8000 München 2
Tel. 0 89/28 12 28



CeBIT - Welt-Centrum
Büro · Information · Telekommunikation

Große Ereignisse werfen bekanntlich ihre Schatten voraus, so auch die größte Messe der Welt in dieser Branche, die CeBIT'88 in Hannover. Damit Sie sich auf der Messe ein wenig besser zurechtfinden, geben wir Ihnen hier ein paar wichtige Informationen vorab:



Messedauer:
Die CeBIT ist von Mittwoch, den 16.03. bis einschl. Mittwoch, den 23.03.88 täglich, auch am Wochenende, von 9 bis 18 Uhr für Besucher geöffnet.

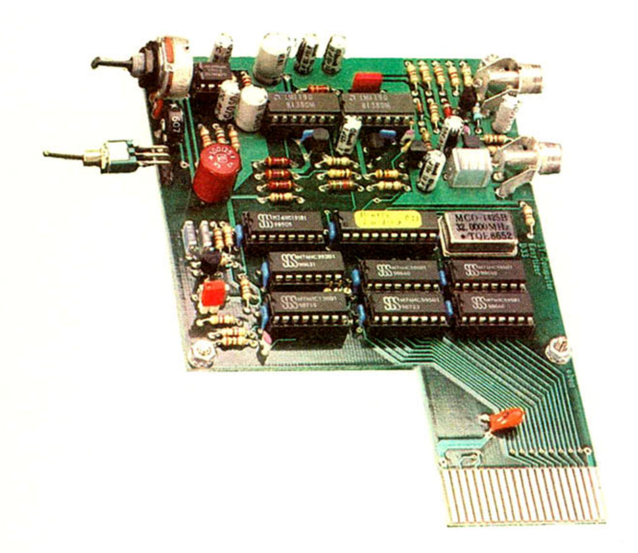
Eintrittskarten:
- Im Vorverkauf bei Industrie- und Handelskammern sowie den Handwerkskammern:
Dauerkarte (für alle acht Tage): DM 48,-
Tageskarte: DM 19,-
- An den Kassen des Messegeländes:
Dauerkarte (für alle acht Tage): DM 58,-
Tageskarte: DM 24,-

Schüler und Studenten erhalten die Tageskarte gegen Vorlage des Schüler- bzw. Studenten-Ausweises für DM 12,-. Jugendlichen unter 14 Jahren ist der Eintritt nicht gestattet.

Der ATARI Stand...
befindet sich in Halle 7 Stand D46-E57 und ist mit 720 m2 Grundfläche der zweitgrößte Stand in dieser Halle. Zu dieser Grundfläche kommen noch ca. 120 m2 Besprechungs- und Konferenzräume im ersten Stock hinzu. An 50 Arbeitstischen wird ATARI zeigen was so alles 1988 auf uns zu kommt. Neben dem neuen Transputer Rechner mit dem Betriebssystem Helios wird neuste Software für die ST Serie gezeigt. Aber auch ATARIs PC-Serie wird mit vier Modellen vertreten sein. Und wer Jack Tramiel kennt, kann sicher sein, daß noch so manch' andere Überraschung zu sehen sein wird.
Und wenn Sie schon auf der CeBIT sind, sollten Sie unbedingt den Stand der MERLIN Computer in Halle 17 Stand A70 besuchen. Dort können Sie Fragen an die Redaktion der ST Computer richten und die neusten Hardware-Projekte wie z.B. den ATARI Monitor SM 124 als "Multisync" für alle Auflösungsstufen betrachten.
Also - auf geht's nach Hannover!

Aus der Hardwareküche





Easytizer
– der Videodigitizer ohne Geheimnisse

Mit dem Easytizer können Sie beliebige Videosignale von einer Schwarzweiß- oder Farb-Kamera, Videorecorder oder auch direkt vom Fernsehgerät (mit Composite Video Ausgang) digitalisieren und somit auf dem Bildschirm Ihres ATARI ST sichtbar machen.

Der Easytizer wird am Modul-Port des ST angeschlossen. Die Auflösung beträgt 800x600 Bildpunkte, so daß in mittlerer Auflösung 640x200 Bildpunkte in vier Graustufen dargestellt werden können. In dieser Betriebsart werden **12,5 Bilder in der Sekunde** wiedergegeben! Im hochauflösenden Modus werden 640x400 Bildpunkte in Schwarzweiß wiedergegeben.

Besondere Merkmale des Easytizers:

- Software vollständig in Assembler
- Abspeichern der Bilder im DEGAS-Format
- Spiegeln eines Bildes in horizontaler und vertikaler Ebene
- Von STAD ansteuerbar
- Ein beliebiger Bildausschnitt kann in ein mit der Maus wählbares Format verkleinert oder vergrößert werden
- Animation mit beliebig vielen Bildern möglich, nur durch die Kapazität des Rechners begrenzt (beim Mega ST4 über 100 Bilder)
- Eingebauter Druckertreiber für die mittlere Auflösung für NEC P6/P7 und EPSON oder Kompatible
- Wahlweise automatische oder manuelle Helligkeitseinstellung
- Schnappschuß

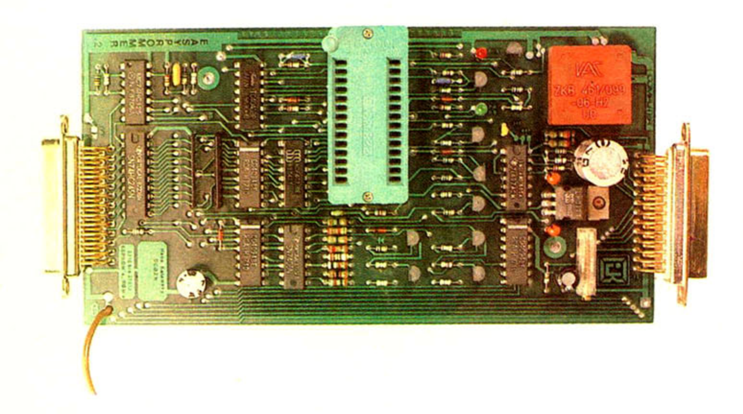
Lieferumfang:

1.) **Fertiggerät**
komplett aufgebaut und geprüft, inklusive Diskette mit der Easytizer Software und Bedienungsanleitung. Preis: DM 289,–

2.) **Teilesatz**
Für Bastler liefern wir einen Teilesatz bestehend aus:
- doppelseitiger, elektronisch geprüfter Platine mit Lötstopplack und Bestückungsaufdruck sowie vergoldeten Anschlußkontakten
- fertig programmiertes GAL 16V8
- Quarzoszillatormodul 32 MHz
- Diskette und Bedienungsanleitung — Preis: DM 129,–

Der Easyprommer V2.2

Ein universelles Programmiergerät für den ATARI ST, das nicht nur alle gängigen EPROMs der 27...-Serie inkl. 27011 (Megabit-Eprom) brennt, sondern auch die modernen EEPROMs lesen und programmieren kann. Die sehr komfortable Software, natürlich voll GEM unterstützt, erlaubt alle nur denkbaren Manipulationen. Mit dem integrierten HEX/ASCII-Monitor lassen sich Änderungen der Daten blitzschnell realisieren. Der auf der Platine befindliche Druckertreiber erlaubt das problemlose Drucken der Daten. Zwei Zusatzprogramme ermöglichen das Brennen **jeder** nicht kopiergeschützten Software direkt von Diskette einschließlich Accessories und Autostart-Programmen. Der Easyprommer wird an den Druckerport angeschlossen und samt Diskette und Handbuch geliefert.



Hiermit bestelle ich

☐ Easyprommer (Fertiggerät) für DM 349,–

☐ Easyprommer (Bausatz) für DM 298,–

☐ Easytizer (Fertiggerät) für DM 289,–

☐ Easytizer (Teilesatz) (w. o. beschrieben) für DM 129,–

☐ 128 KByte EPROM-Karte (fertig bestückt o. EPROMs) für DM 58,–

☐ Verlängerungskabel für Easyprommer für DM 39,90

Versandkosten: Inland DM 7,50 Ausland DM 10,–
Auslandsbestellungen **nur** gegen Vorauskasse

☐ Vorauskasse
☐ Nachnahme

Name: ____________ Vorname: ____________

Straße: ____________ Ort: ____________

Unterschrift: ____________

Bezugsquelle:

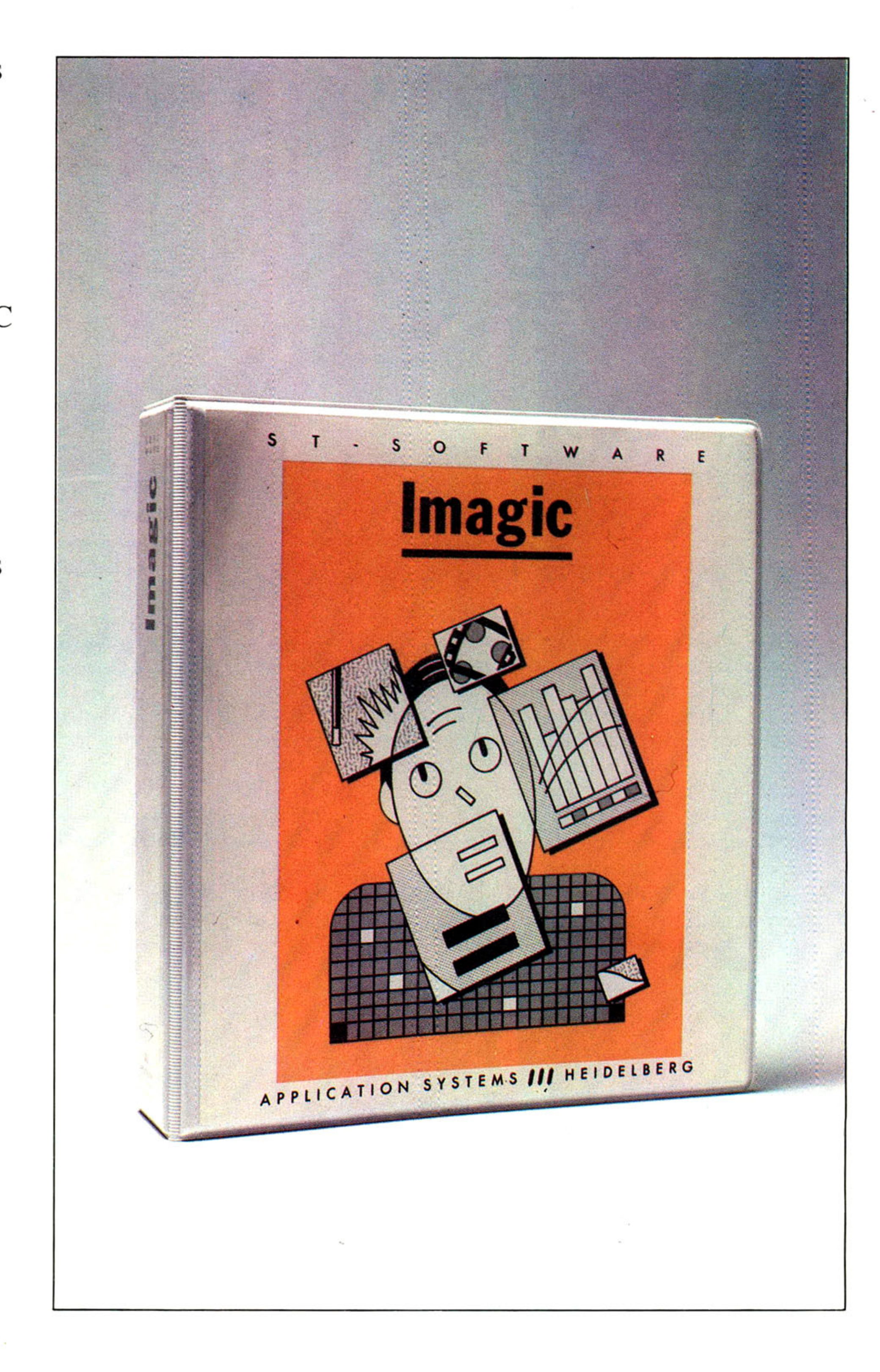


Industriestraße 26
6236 Eschborn

ALS DIE BILDER

IMAGIC, Grafik

Spätestens seit der CeBit des Jahres 1987 kennt man den Begriff IMAGIC aus diversen Zeitschriften. Obwohl immer wieder im Zusammenhang mit IMAGIC die unterschiedlichsten Grafik-Demos auftauchen und auch der Begriff Grafik-Compiler gefallen ist, herrscht Verwirrung, um was es sich tatsächlich handelt. Nachdem sich vor ein paar Monaten das Geheimnis des Distributors zu Gunsten von APPLICATION SYSTEMS /// HEIDELBERG gelichtet hat, durften wir die Autoren von IMAGIC besuchen, um das Geheimnis des Programms zu lüften.



LAUFEN LERNTEN...

ohne Grenzen!?

Um es vorwegzunehmen, ein Programmpaket oder besser ein Entwicklungssystem, um das es sich eindeutig handelt, gibt es in dieser Art bisher auf einem Rechner der Preisklasse des ST noch nicht.
Was ist IMAGIC ? Ein Zeichenprogramm, ein Compiler, ein Interpreter, eine Bildverarbeitung ? Von allem etwas! Verwirrt ? Nicht gleich verzweifeln, wir werden das Konzept schon erläutern. Zunächst stellt sich bei der Betrachtung eines solch komplexen Paketes die Frage nach dem Ziel oder der Aufgabe. Nach einem Interview mit den Programmautoren und der Beschäftigung mit IMAGIC läßt sich diese Frage eigentlich relativ deutlich beantworten. Ziel dieses Programmpaketes ist es, eine Demonstration beliebiger Bilderfolgen oder Animationen zu erstellen, die auf die vielfältigste Weise - es gibt zum Beispiel Überblendeffekte oder ähnliches - verfeinert werden, so daß sie dem Betrachter angenehm auffällt. Beispiele wären hier Trickfilme, Händlerdemonstrationen und vieles mehr. Soweit erst einmal eine Vorstellung des Endproduktes, das mit IMAGIC erarbeitet werden soll.

geht weiter...

Grafiken sind etwas für das Auge

In der Anleitung heißt es: "Das Programmpaket IMAGIC ist ein grenzenloses Werkzeug für die Arbeit mit Grafiken. Es stellt Funktionen bereit für das Erstellen, Bearbeiten, Verknüpfen und Präsentieren von Bildern." Dieses Programmpaket läßt sich besser in fünf verschiedene Werkzeuge unterteilen, die wir im folgenden beschreiben wollen. Zunächst sollte aber noch ein wenig auf das Konzept eingegangen werden. Wie oben schon erwähnt, ist Ziel des Arbeitens das Erstellen einer Show von Grafiken, die mit Effekten versehen worden ist. Da Effekte von Veränderungen ihrer Parameter leben, lag es für die Autoren nahe, eine Sprache zu entwickeln, die erstens leicht erlernbar - daher mußte es zwangsläufig ähnlich einer Hochsprache wie zum Beispiel PASCAL werden - und zweitens, für das Erstellen einer Show, möglichst vielfältig ist. Diese PASCAL-ähnliche Sprache wird nun dazu verwendet, anhand der gegebenen Bilder eine ansprechende Show zu erstellen. Programmiert wird, wie man es bei anderen Compilersprachen gewöhnt ist, in einem Editor, worauf der Sourcecode später von einem Compiler in einen Code umgesetzt wird. Der IMAGIC-Compiler erzeugt aus dem IMAGINE-Quellkode (IMAGINE ist der Name der Sprache) einen Kode, der von einem RUN-TIME-Modul verarbeitet werden kann (Bild 1). Dadurch können die kompilierten Shows durch Weitergabe der Show und dieses Moduls ohne das eigentliche Programmpaket laufen- die Weitergabe des RUN-TIME-Moduls ist natürlich erlaubt. Vielfach ist es aber der Fall, daß eine Person gerne eine Show erstellen möchte, aber bisher noch nie programmiert hat. Dem Manne (der Frau) kann geholfen werden, denn im Paket ist auch ein Programmgenerator integriert, mit dem man auf grafische Weise Shows erstellen kann, die dann als IMAGINE-Quellkode abgespeichert werden können. Da dies etwas viel auf einmal war, wollen wir die einzelnen Werkzeuge etwas genauer unter die Lupe nehmen.

Ein Rendezvous mit DENISE

Unter dem Namen DENISE verbergen sich drei Dinge: Erstens ein Zeichenprogramm, zweitens Möglichkeiten der Bildverarbeitung und drittens der Grafikinterpreter.

Tausend und ein Bild

"Schon wieder ein Zeichenprogramm !", wird der ein oder andere Leser sicherlich sagen, aber es ist wichtig, daß in einem solchen Paket ein leistungsfähiges Zeichenprogramm vorhanden ist. Im allgemeinen wird man zwar mit einem - in dem Paket aber nicht enthaltenen - Digitizer Bilder digitalisieren, trotzdem müssen diese dann in den meisten Fällen noch bearbeitet werden. Außerdem kann eine Show nicht nur aus digitalisierten Bildern bestehen, sondern es werden immer einige eigene Zeichnungen dabei sein. Erhebt IMAGIC einen professionellen Anspruch, muß auch ein professionelles Zeichenprogramm vorhanden sein. Begrüßt wird man in DENISE mit der in Bild 2 abgebildeteten Erscheinung, die zunächst einmal das Hauptmenü sowie einen Auschnitt der Bilddatenbank darstellt, in der die Bilder

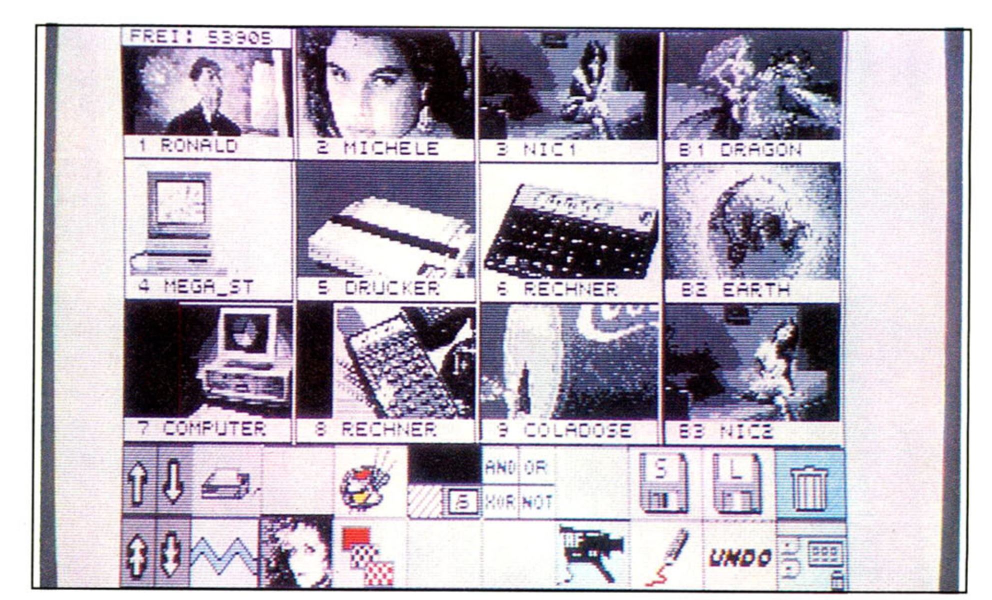


Bild 2: MENUEPIC

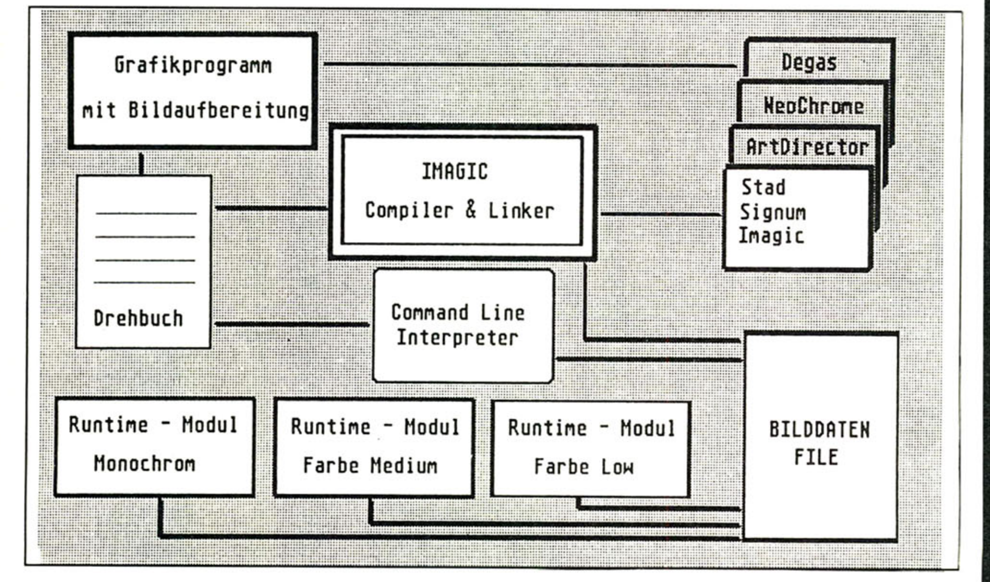


Bild 1: Das Konzept von IMAGIC

verkleinert und in Schwarz/Weiß gezeigt werden. Von anderen Zeichenprogrammen ist man bereits gewohnt, daß man mehrere Bilder gleichzeitig verarbeiten kann; bei DENISE können bis 1001 Bilder zur selben Zeit im Speicher sein, denn fast alle Bilder - bis auf drei besondere - liegen 'gepackt' im Speicher. Das bedeutet, daß die Daten des Bildes untersucht und auf raffinierte Weise zusammengefaßt werden, so daß sie weniger Speicherplatz verbrauchen als normal. Es ist einsichtig, daß das Packen von Bildern aufgrund verschiedener Verfahren durchführbar ist, wobei die Firma IMAGIC Grafik nicht weniger als ZWANZIG verschiedene Verfahren implementiert hat. Zugegeben, das Packen benötigt seine Zeit - das Auseinanderpacken des Bildes ist allerdings unvergleichbar schneller -, aber möchte man möglichst viele Bilder im Rechner, ist man gerne bereit, diese Zeit in Anspruch zu nehmen. Erwähnenswert ist, daß DENISE alle gängigen Bildformate wie DEGAS (gepackt), DOODLE, STAD, Art Director, SIGNUM, NeoChrome und sogar das AMIGA IFF-Format laden kann. Zu erwähnen ist auch, daß DENISE in allen Grafikstufen arbeitet und die geladenen Bilder in die jeweilige Grafikstufe konvertiert!

Die Farbauswahl stellt sich, wie auch von anderen Programmen bekannt, in einer Farbpalette von 64 Farben dar, denen in 8 Stufen Blau zugemischt werden kann, so daß die bekannten 512 Farben des ST erreichbar sind. Trotzdem sind nur Bilder mit maximal 16 Farben gleichzeitig möglich. Zugegeben, es gibt heute Programme auf dem ST, die es ermöglichen, mehr Farben gleichzeitig darzustellen, bedenkt man aber, daß dies ein Großteil der Rechenzeit des Computers verschlingt, ist es nicht mehr möglich, Effekte und sogar Animationen zu erzeugen- wofür hat man sich aber dann die Mühe gemacht, solche Bilder zu zeichnen ?. Eine Farbvielfalt mit mehr als 512 Farben zu erzeugen, wäre zwar machbar, meinen die Autoren von IMAGIC, sie ist aber dem Betrachter aufgrund des flackernden Bildschirms nicht mehr zumutbar. Eine interessante Möglichkeit bietet das Anpassen von Farbpaletten zweier Bilder. Dabei werden in beiden Bildern durch ein Verfahren nicht einfach Farben ausgetauscht, sondern es werden ähnliche Farben gesucht und diese beiden Bildern angepaßt- das Ergebnis ist verblüffend und interessant. Diese Funktion des Programmes ist wichtig für die Erstellung einer Demo, denn sollen zwei farbige Bilder ineinander übergeblendet werden, so ist dies nur möglich, wenn sie die gleichen Farbpaletten besitzen. Natürlich sind die sonst üblichen Möglichkeiten, die ein Zeichenprogramm bietet, wie Punkt, Linie, Mehrfach-Linie, Kreis, Rechteck und so weiter, vorhanden. Dabei kann man sogar ein sogenanntes Quellbild definieren - dies kann zum Beispiel ein Ausschnitt eines Bildes sein - und mit diesem dann zeichnen oder es etwa als Füllmuster benutzen.



Bild 3: Ronny

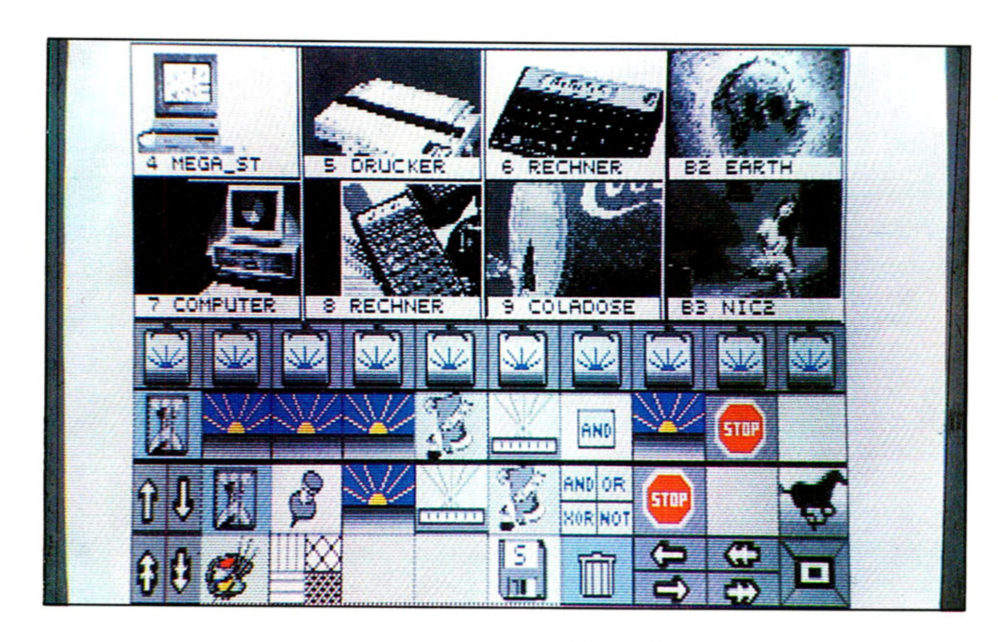
Wie eine Folie

Dieses Quellbild kann aber auch als Hintergrund benutzt werden, auf dem gezeichnet wird- das bedeutet, man kann zum Beispiel eine Sequenz eines Trickfilms zeichnen. Die nächste Sequenz wird sich aber nur geringfügig ändern, so daß man das letzte Bild einblendet und auf dieser Unterlage die neue Sequenz zeichnet; ist die Zeichenarbeit beendet, wird das vorherige Bild einfach wieder weggeblendet- eine sehr nützliche Technik, zumal das Ein- und Ausblenden über eine Taste schnell vonstatten geht.

Ein weiteres erwähnenswertes Feature ist der Mapper, der das beliebige Verziehen eines Objektes anhand einer Maske ermöglicht. Benutzt man beispielsweise eine Maske in Form eines Kreises, stellt sich DENISE diesen als dreidimensionales Gebilde, also als Kugeloberfläche vor. Nun kann man ein frei wählbares Bild - oder auch einen Bildauschnitt - auf diese Kugeloberfläche projizieren lassen. Der Kreativität sind keine Grenzen gesetzt (Bild 3). Weitere erwähnenswerte Funktionen wie Textausgabe mit Blocksatz, stufenloses (!) Vergrößern und Verkleinern von Bildausschnitten, Spiegeln an verschiedenen Achsen und gradweises Drehen sind gut in die Praxis umgesetzt worden, wobei mich besonders das Verkleinern angenehm überrascht hat, da selbst auf sehr kleinen Bildern noch das eigentliche Bild erkennbar ist.

DENISE - Die Bildverarbeitung

Bildverarbeitung wird dazu benutzt, um meist digitalisierte Bilder rechnerisch umzuarbeiten, wobei diese Rechnungen auch von einer Elektronik übernommen werden können. Bei der Digitalisierung von Bildern taucht zum Beispiel häufig der Fall auf, daß das Bild 'verschneit' ist. Diese Punkte kann man dann rechnerisch herausfiltern. Andere Möglichkeiten sind das Hervorheben von bestimmten Grau(Farb)-Stufen oder vielleicht Kanten oder das Kontrastieren eines Bildes. Oftmals ist sogar ein besseres Erkennen unterschiedlicher Details auf vormals nebligen und scheinbar unscharfen Bildern nach der rechnerischen Bearbeitung möglich. Natürlich kann diese Bearbeitung keine Details herzaubern, wo keine sind und die Handhabung erfordert ein wenig Geschick und Erfahrung, trotzdem lassen sich nach einiger Übung beachtliche Erfolge erzielen (Bild 4).



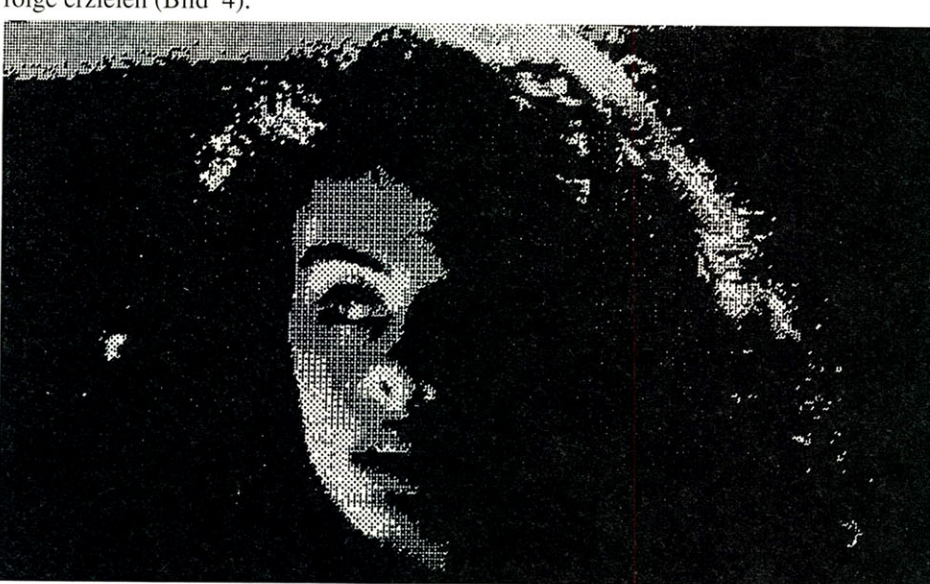


Bild 4.1: Vor und...



Bild 4.2: ...nach der elektronischen Bearbeitung

DENISE - der Grafikinterpreter

Wie schon oben erwähnt, kann eine Show auch auf grafische Art und Weise produziert werden. Diese Art ist sicherlich die angenehmste und einfachste, da sie es ermöglicht, daß selbst programmierunerfahrene Anwender ihre Shows erstellen können. Da dieser Teil einen Programmkode in normalem IMAGINE-Programmtext erstellt, kann man anhand der erstellten Demo und des erzeugten Programms sehr gut die Sprache IMAGINE erlernen. Es handelt sich also um einen Programmgenerator. Das weitere Verfahren zur Erstellung einer eigenständigen Show geht dann wie oben erwähnt: Kompilieren, Linken (dabei werden die verwendeten Bilder komprimiert und zusammengefaßt) und Starten.

Der Grafikinterpreter ist ähnlich einem Storyboard aufgebaut (Bild 5). In jedes Feld kann man Icons unterschiedlicher Bedeutung legen. Durch Selektieren eines beliebigen Bildes aus der Bilddatenbank und eines Icons wird ein Bild mit einem Überblendeffekt verbunden. Eine sich daraufhin öffnende Box gibt eine sinnvoll maus- und tastaturunterstützte Eingabe der Parameter frei. Wichtig ist, daß diese Parameter später auch immer wieder geändert werden können. Weitere Icons bieten den Komfort, eine Pause zwischen Bildern oder Parametern, wie Spiegelung oder Neuzeichnen, einstellen zu können.

Die Arbeitsweise ist sehr einfach und benutzerfreundlich, daher wird man sehr schnell lernen, damit umzugehen. In relativ kurzer Zeit hat man einen Film erstellt, den man auch sofort in seiner Art und Weise betrachten kann, so daß der sogenannte 'AHA'-Effekt beim Lernen um einiges

größer ist, als bei dem Arbeiten mit dem Compiler. Hervorragend ist, daß selbst die Schleifenstruktur in DENISE verwirklicht worden ist. Dadurch ist es möglich, bei genügend Speicherplatz, eine gesamte Show in DENISE zu verwirklichen.

Die Programmiersprache IMAGINE

Wie anfangs schon erwähnt, ist IMAGINE eine PASCAL-ähnliche Programmiersprache, die zur Erstellung von Bilderdemonstrationen erfunden und in die Praxis umgesetzt worden ist. IMAGINE schreibt dem Programmierer - wie in PASCAL - eine gewisse Struktur vor (Vereinbarungsteil, Programmteil ...) und bietet ihm von höheren Programmiersprachen gewohnte Schleifenstrukturen. Da ein Grafikinterpreter vorhanden ist, der Quelltexte erzeugt, ist ein Erlernen dieser ohnehin relativ einfachen Sprache, ganz besonders dann, wenn man PASCAL oder C beherrscht, sehr einfach. IMAGINE bietet zusätzlich einige sehr interessante Dinge, die im folgenden erwähnt werden sollen.

Mehr als 71 Bilder pro Sekunde sind nicht sinnvoll

Durch die hohe Packungsdichte und die sehr stark optimierte Speicherverwaltung ist es möglich, circa 1000 Bilder im Speicher zu halten und diese selbst im gepackten Format mit bis zu 25 Bildern pro Sekunde ablaufen zu lassen. Natürlich ist eine enorme Geschwindigkeitssteigerung vorhanden, wenn die Bilder ungepackt im Speicher liegen! Dazu ein Wort: Theoretisch wären auch höhere Geschwindigkeiten als 71 Bilder pro Sekunde für ungepackte Bilder möglich, aber da der ATARI ST im Monochrom-Modus nur 71 Bilder pro Sekunde - in Farbe sind es höchstens 60 - darstellen kann, wäre eine höhere Geschwindigkeit nicht sinnvoll gewesen. In IMAGIC wird, um ein Flakkern zu vermeiden, auch dafür gesorgt, daß die Bilder nur dann umgeschaltet werden, wenn der Computer das Bild am Monitor gerade nicht aufbaut. Alle Angaben, daß ein Programm mehr als 71 Bilder pro Sekunde darstellen könne, sind unseriös, denn es ist zwar möglich, mehr Bilder pro Sekunde im Speicher aufzubauen, aber unmöglich, diese höhere Anzahl auch anzuzeigen.

Eine weitere, besondere Möglichkeit ist das Abfragen des Zehnerblocks von IMAGINE aus. Dadurch kann tastenabhängig in bestimmte Bereiche der Demonstration verzweigt werden. Eine mögliche Anwendung wäre beispielsweise das Erstellen einer Demonstration im Lehrbereich, wobei über Tastatur bestimmte Wissensgebiete ausgewählt werden können.

IMAGINE unterhält sich über MIDI

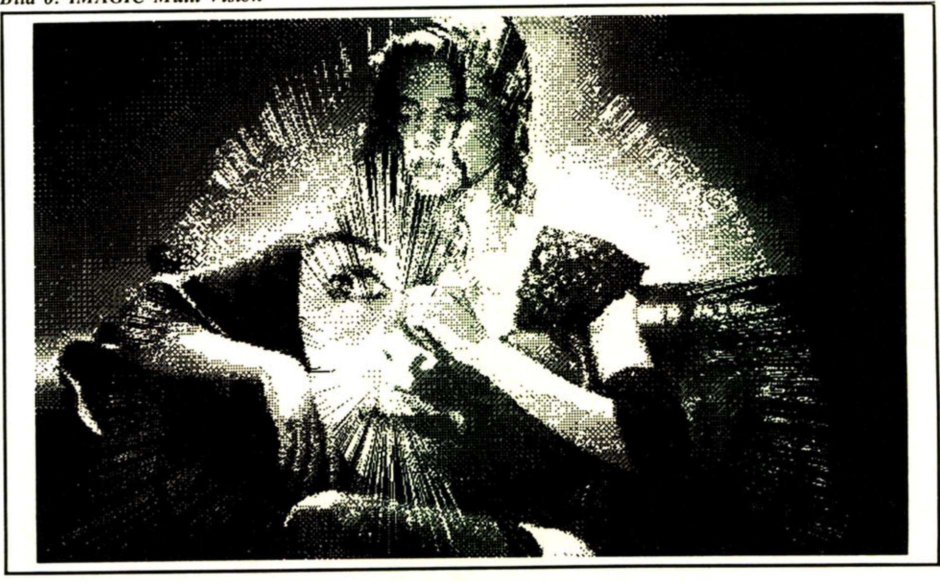
Last but not least soll noch die Fähigkeit von IMAGINE erwähnt werden, Kommandos über MIDI an andere Rechner weiterzugeben. Durch diese Fähigkeit ist es möglich, bis zu 255 Rechner - eine Möglichkeit, die höchstens die Firma ATARI ausnützen könnte - miteinander zu koppeln, die Daten austauschen können. Auf der Düsseldorfer Messe war ein Beispiel zu sehen: Auf vier Rechnern, die untereinander synchronisiert waren, lief eine Demonstration so ab, daß die vier Monitoren ein Gesamtbild ergaben (Bild 6)! Um dieses Feature ausnutzen zu können, braucht man eigentlich nur ein wenig Fantasie.

Der CLI - Command Line Interpreter

Der CLI ist ein Programm, das in TOS-Umgebung geschrieben worden ist und das dem Ausprobieren von Überblendeffekten (Bild 7) und deren Parametern gewidmet ist. Es ist sicherlich eher für erfahrene Programmierer als für unbedarfte Anfänger geschrieben worden, denn die Benutzung ist aufgrund der kaum vorhandenen Maussteuerung nicht so spielerisch einfach, wie die der mausgeführten Programme. Trotzdem werden bestimmte Anwender ganz sicher ihren Spaß an diesem Programm haben. Die Aufgabe des CLIs ist, dem Entwickler beim Debuggen, also Fehlersuchen, und Verbessern seiner Show zu helfen. Der CLI, der selbst sehr wenig Platz im Speicher des ST benötigt,



Bild 6: IMAGIC Multi-Vision


Bild 7: Überblendung von Bildern

ist in der Lage, auch relativ große Shows laden zu können, bedenkt man, daß DENISE mit allem Drum und Dran etwa 600 Kilobytes verbraucht. Beim Debuggen lädt man die Show und läßt sie bis zu dem Punkt laufen, an dem man einen 'Fehler' findet. Effekte lassen sich dann im CLI ausprobieren und austesten.

Meister des Schnappens

Der Snapmaster ist eine weitere Zugabe in IMAGIC und dient dem Abspeichern von Bildern aus beliebigen anderen Programmen. Vielfach gibt es Programme, die es ermöglichen, hervorragende Grafiken oder Sequenzen zu erstellen, lassen ein Abspeichern zum Beispiel von Einzelbildern einer Sequenz aber nicht zu. Um diese Bilder aber doch in IMAGIC verwenden zu können, ist der Snapmaster - einmal installiert - immer über Knopfdruck zu erreichen. Interessant ist, daß man den Rechner so manipulieren kann, daß er nach jedem Bildschirmaufbau anhält und ein Bild - einer Sequenz - durch Snapmaster abspeichert. Bisher ist es mir noch aus jedem Programm, das Bilder beinhaltete, die mir gefielen, gelungen, diese auch "herauszuschnappen.

Vor- und Abspann für Videofilme

Als Zielgruppen kann man einmal den Hobbybereich anführen, der in erster Linie den Anwender anspricht, der Spaß am Erstellen von Demos hat oder ihn für seinen Heim-Video-Bereich nutzen möchte. Erwirbt man beispielsweise ein GENLOCK-Interface, so ist es möglich, Grafiken, einen Vorspann oder Abspann für die eigenen Videofilme zu erstellen und einzublenden. Damit ergeben sich ganz neue Möglichkeiten für den Heim-Video-Bereich. Die andere Zielgruppe wird im professionellen Bereich der Erstellung von Demos -in Zusammenhang mit einem Digitizer wie beispielsweise dem EASYTIZER- zur Präsentation neuer Produkte in Schaufenstern für Händler zu finden sein.

Resümee

Die Benutzerführung von IMAGIC ist durch die Verwendung von symbolischen Menüs sehr einfach zu verstehen, und der Rechner ist so gut programmiert, daß der Benutzer in den seltensten Fällen - höchstens bei aufwendigen Berechnungen - auf ihn wartet. Eine besondere und nicht selbstverständliche Tatsache ist, daß das Zeichenprogramm mit all seinen Funktionen in allen Auflösungen arbeitet. Da das Erstellen eines IMAGINE-Programms fast vollständig durch das Programm DENISE durchführbar ist, ist ein Einarbeiten in IMAGINE und damit das Kreieren von eigenen Demos selbst für Anfänger sehr leicht zu erlernen.
Alles in allem erkauft man sich mit IMAGIC ein Grafik-Entwicklungssystem, das seinesgleichen auf dem ATARI ST sucht und bisher mit einer solchen Vielfalt an Möglichkeiten noch nicht verfügbar war. Wem also 498,- DM für ein Grafikentwicklungssystem nicht zu viel sind, dem kann sicherlich zu dem durchdachten Paket IMAGIC von Application Systems /// Heidelberg geraten werden. Ob er damit professionelle Shows erstellen möchte, mit denen er im allgemeinen den Anschaffungspreis schnell wieder verdient hat, oder ob er nur Spaß am Erstellen von Demos hat oder vielleicht seine eigenen Videos ver-schönern möchte, bleibt ihm freigestellt.

Stefan Höhn

Bezugsadresse: Application Systems /// Heidelberg
Englerstr. 3
6900 Heidelberg
Tel.: 06221/300002



Interview mit Alexander Beller und Jörg Drücker der Firma IMAGIC GRAFIK:

ST: Wann habt Ihr beide angefangen zu programmieren und was habt Ihr vorher gemacht ?
Alexander Beller: Angefangen hat es bei mir mit dem ZX81 und VC20; den hatte ich vier Wochen, dann waren die Programme zu lang und der Speicher zu klein. Dann habe ich mir einen 64er gekauft und darauf programmiert, obgleich nichts dabei herauskam, was man verkaufen hätte können- an so etwas habe ich damals noch nicht gedacht.
ST: Warst Du damals schon besonders an Sound und Grafik interessiert ?
Alexander Beller: Nein, eigentlich nicht! Ich habe nur ein wenig das BASIC des C64 erweitert. Mit BASIC selbst habe ich damals erst garnicht richtig angefangen, sondern praktisch alles in Assembler geschrieben. Später, als es dann einen ATARI ST gab, habe ich noch überlegt, ob ich mir einen AMIGA kaufen soll, aber ein Bekannter hat einen ATARI und den AMIGA gab es noch nicht - also habe ich mir einen ATARI gekauft; das ist jetzt etwa eineinhalb bis zwei Jahre her.
ST:Und was hast Du dann mit dem ST gemacht ? Gleich mit IMAGIC angefangen ?
Alexander Beller: Nein, nein. Ich hatte ja nur ein Spiel (Brattacas) und das ATARI BASIC, in das ich fünf Zeilen eintippte. Dieses Programm lief dann nicht, und daraufhin war ich überzeugt, weiterhin Assembler zu programmieren.
ST: Wie habt Ihr Euch kennengelernt ?
Alexander Beller: Über einen Bekannten habe ich Jörg kennengelernt, mit dem wir Bilder digitalisierten und Demos zusammenstellten.
ST: Und wie fing die ganze Sache bei Dir an, Jörg ?
Jörg Drücker: Ich habe vor circa 10 Jahren auf einem Großrechner während meines Informatikstudiums angefangen, in PASCAL und Assembler zu programmieren. Bis auf einen 8Bit-ATARI, den ich 2-3 Wochen mal in den Fingern hatte, kannte ich bis dahin eigentlich keinen kleinen Rechner. Später überlegte ich mir dann, einen Rechner für die Buchhaltung und Textverarbeitung zu kaufen. Da der ST am Anfang sehr teuer war, zögerte ich noch eine Weile, kaufte ihn mir dann aber doch. Die erste Zeit spielte ich mit dem Rechner, allerdings befriedigte das nicht besonders, so daß Alex und ich anfingen, die ersten Demos zu schreiben. Das waren dann

WISSEN SIE, WAS IN IST?

C&C
Computers and Communications

In ist ein Monitor, der zeigt, was der Computer kann.

... einer mit zukunftssicherer MultiSync-Technologie.

In ist auch ein Monitor mit vielen Farb- und Grautönen für brillante Texte und Bilder.

... der außerdem die Augen verwöhnt.

Arbeiten mit MultiSync-Monitoren von NEC ist **in**.

Und **out** ist der triste Bildschirm-Alltag mit antiquierten Monitoren. Heute und erst recht in Zukunft.





Die Multitalente für Text, Business Grafik, CAD/CAM und Desktop Publishing:



MultiSync II
14" 800 x 560
Color, TTL/
Analog



MultiSync Plus
15", 960 x 720
Color, TTL/
Analog



MultiSync GS
14", 900 x 700
Monochrom,
TTL/Analog



MultiSync XL
20", 1024 x 768
Color, TTL/
Analog

MultiSync-Monitore wurden an folgenden PC's getestet:

Apple, Atari, Commodore, Compaq, Epson, Fujitsu, Hewlett Packard, IBM, Kaypro, Multitech, Nixdorf, Olivetti, Panasonic, Plantron, Sanyo, Schneider, Siemens, Tandon, Toshiba, Victor, Zenith.



NEC Deutschland GmbH

Klausenburger Straße 4
8000 München 80 West Germany
Tel.: 089/93006-0
Telefax: 089/937776/8
Telex: 5218073 und 5218074 nec m

irgendwelche Grafiken, die wir zu einer Show zusammenstellten. Dabei bemerkten wir, daß es wohl interessanter sei, bestimmte Effekte einzubauen. Nach einer Weile hatten wir einige verrückte und interessante Effekte in Assembler geschrieben. Dies führte allerdings dazu, daß wir diese in nächtelanger Arbeit irgendwie "zusammenstrickten". Irgendwie haben wir dies dann zum Laufen gebracht. Es gab zu diesem Zeitpunkt weder eine Sprache noch ein Konzept. Man kann also sagen, die Sache ist spielerisch entstanden.

Alexander Beller: Da uns bei längeren Shows auffiel, daß es lästig ist, die Assembler-Sources immer zusammenzuwürfeln, begannen Jörg und ich, die Sprache von IMAGIC in stundenlangen Telefongesprächen zu entwickeln.

ST: Wie würdet ihr die Aufgabe vonIMAGIC beschreiben Jörg Drücker: Man hat die Aufgabe,Grafiken auf dem Rechner möglichst anspruchsvoll darzustellen. Wenn man anfängt, Effekte zu schreiben, wird man merken, daß die Effekte von den Parametern leben. Oft sind es dann fünf oder zehn und bei einer Eingabe mit der Maus war dann ganz schnell Schluß, so daß der Gedanke an eine Sprache kam. Anfangs war die Sprache Assembler sehr ähnlich - das kann man aber keinem zumuten. Also überlegten wir uns, daß es eine Hochsprache werden sollte, und wenn es eine Hochsprache sein sollte, dann möglichst PASCAL-ähnlich.

Zusammengefaßt kann man sagen: Die Aufgabe von IMAGIC ist es, Grafiken Leben einzuhauchen und dies auf eine möglichst benutzerfreundliche und vielfältige Art, das heißt Trickfilme, Überblendeffekte, schnelle Effekte jeder Art...

ST: Also ähnlich wie im Fernsehen..

Alexander Beller: Richtig. Vorspann, Abspann, Werbegrafik und so weiter. Eigentlich war ja die Grundidee, die besten Demos zu machen, die auf dem ATARI ST irgendwie möglich sind - nicht nur so ein Ball, der hoch und runter springt, sondern auch anspruchsvollere oder lustige Demos. Anfangs wollten wir das Programm nicht verkaufen, nur haben wir später gemerkt, daß für diese Art Programme ein Markt vorhanden ist.

ST: Das bedeutet, Ihr habt ein Konzept erarbeitet, das die Möglichkeit bietet, daß Personen, die Programmiererfahrung haben, mit einer Hochsprache ihre Demos und Trickfilme erarbeiten können, während reine Anwender anhand einer grafischen Oberfläche ihre Demos "spielerisch" oder besser benutzerfreundlich erstellen können ?!

Jörg Drücker: Man kann praktisch IMAGIC in zwei Teile teilen: Ein Teil ist recht maschinennah als Hochsprache zugänglich, während der andere sehr anwenderfreundlich ist, so daß man Demos erstellen kann, ohne Ahnung von Algorithmen zu haben. Hier schließt sich der Kreis wieder, denn ganz zum Schluß steht eigentlich doch eine optionale Mauseingabe, die eine Show erstellt, die vom Compiler verarbeitet werden kann.

ST: Also ist diese Grafikoberfläche eine Art Programmgenerator ?!

Jörg Drücker: So ist es. Dieser Teil von IMAGIC generiert tatsächlich einen Quellkode, der auch nachträglich mit der Hand überarbeitet werden kann. IMAGIC kann man als große Bibliothek von Grafikeffekten und neuerdings auch grafischen Bild verarbeitungstechniken betrachten. Aus diesem großen Topf greift man die gewünschten Teile heraus, manipuliert sie und findet das Ergebnis hoffentlich schön. Zu alledem hat man noch die Möglichkeit, Bilder mit Hilfe eines Grafiktools zu verändern und zu bearbeiten und zu einem Vorspann, Trickfilm usw. zusammenzustellen. Eine gute Demo zu schreiben, heißt normalerweise, in Assembler zu programmieren und das kann man keinem Anwender zumuten.

ST: Wen seht Ihr denn als Zielgruppe des Programms ?

Jörg Drücker: Wie sehen als Zielgruppe zum Beispiel Händler, die sich zu einem Preis von 900-2000 DM eine Leuchtschrift kaufen, die nichts anderes macht, als eine einfallslose Laufschrift darzustellen. Wer sich so etwas kauft, erwirbt vielleicht für etwas weniger Geld einen ATARI und das Programm IMAGIC und dadurch eine viel anspruchsvollere Werbemöglichkeit !

Er kann diesen Werbeträger einmal herstellen und beispielsweise an alle anderen Filialen verteilen, die diesen ohne das Programm IMAGIC ablaufen lassen können, denn die Demos benötigen IMAGIC nicht mehr.

ST: Wenn ein Käufer mit Eurem Programm eine Demo erstellt, muß er dann noch eine Lizenzgebühr bezahlen ?

Jörg Drücker: Die Benutzung ist völlig frei - er muß nur die Registrationskarte einschicken und damit hat er automatisch das Recht, die Demos weiterzugeben und zu verkaufen.

ST: Ihr seht also die Zielgruppe im Werbebereich?

Alexander Beller: Sicherlich ist die Hauptzielgruppe die Werbung, aber als wir angefangen haben, hat es uns schlichtweg Spaß gemacht, Demos zu erstellen. Vielleicht gibt es Leute, die gern zeichnen oder anderweitig kreativ sind und die IMAGIC benutzen können.

ST: Gibt es eine Möglichkeit, die Routinen, die IMAGIC zur Verfügung stellt, in eigenen Programmen zu benutzen ?

Jörg Drücker: Ja. Es ist möglich, das Run-Time-Modul von einem Programm aus zu laden und dieses dann aufzurufen, um diverse Routinen zu benutzen. Dadurch erweitert sich der Anwenderkreis um ein Vielfaches, da auch die Programmierer, die unsere Grafikeffekte gut finden, diese in ihren Programmen verwenden können. Natürlich ist es damit nicht erlaubt, ein Programm wie IMAGIC zu schreiben...Vorteilhaft ist auch, daß das Run-Time-Modul nur ca. 30Kbytes hat, so daß es bei einem Megabyte-Rechner praktisch nicht ins Gewicht fällt.

Alexander Beller: Für andere Softwarehäuser ist es auch sehr vorteilhaft, ihre Bilder und Animationen zu komprimieren und mit diversen Effekten ablaufen zu lassen, ohne großartig Routinen dafür entwickeln zu müssen.

ST: Wie habt Ihr Euch den Support vorgestellt ? Schließlich ist IMAGIC ein Entwicklungssystem.

Jörg Drücker: Bei einem Preis von 498,- DM ist selbstverständlich ein Service dabei. Das heißt, wenn der Kunde irgendwelche Fragen hat und die Registrationskarte eingeschickt hat, kann er uns um Hilfe bitten.

ST: Das ganze IMAGIC-Konzept klingt sehr interessant. Ist eine Umsetzung auch auf andere Rechner geplant ?

Alexander Beller: Momentan nicht. Wir wollen erst einmal schauen, ob dieses Konzept beim Publikum ankommt. Eins wissen wir relativ sicher, daß eine Umsetzung auf den AMIGA nicht kommen wird. Wahrscheinlicher wäre eine Umsetzung auf neue grafikstärkere Rechner der Firma ATARI.

ST: Das heißt, daß Ihr einen Rechner braucht, auf dem es Euch Spaß macht zu programmieren und das IMAGIC aus Freude am Programmieren entstanden ist, obgleich dahinter ein professionelles Konzept steht?

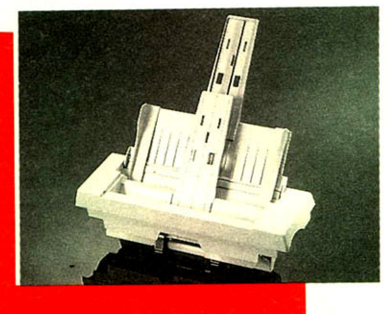
Jörg Drücker: Richtig. Als wir 1987 auf der CEBIT waren, hätten wir nicht gedacht, daß wir ein solches Aufsehen erregen. Zu diesem Zeitpunkt dachten wir zum ersten Mal daran, das Programm zu verkaufen.

ST: Wir danken Euch für dieses Gespräch.

P 2200 – DAS PREIS-LEISTUNGS-GENIE

Die Computer-Anwender haben Grund zum Jubeln!

Genial – endlich ein Drucker, der für Einsteiger, Aufsteiger und Semiprofis geeignet und vor allem erschwinglich ist. Denn NEC erschließt Ihnen jetzt die





Vielseitige, durchdachte Papierzuführungen.

modernste 24-Nadel-Drucktechnologie zu einem erstaunlich günstigen Preis.

NEC Pinwriter P 2200 mit automatischer Einzelblattführung

Erleben Sie eine neue Dimension: gestochen scharfen Korrespondenzdruck mit ungewöhnlich reicher Schriftartenauswahl, brillante Grafik-Darstellung, bequeme Druckersteuerung und integrierte Papierzuführungen.

Warum also tief in die Taschen greifen, wenn es schon für wenig Geld 24-Nadel-Technologie mit allen Raffinessen gibt?

PROFIQUALITÄT ZUM AMATEURPREIS

NEC ist mit seinen 24-Nadel-Druckern in Deutschland marktführend.

Was den P 2200 als echten Profi auszeichnet, sind seine hohe Auflösung von 360 x 360 dpi, ein halbes Dutzend serienmäßiger Schriftarten und eine Reihe praktischer Papierzuführungen. Zum Beispiel können Sie zwischendurch einen Brief drucken, ohne daß das Endlospapier extra herausgenommen werden muß.

Durch seine volle Kompatibilität mit den NEC Pinwritern der 24-Nadel-Serie harmoniert der P 2200 mit allen wichtigen Software-Paketen.

EIN NEC DRUCKER FÜR JEDERMANN

Endlich braucht niemand mehr auf die bewährte NEC Produkt- und Druckqualität zu verzichten.

Hervorragende Druckqualität durch bewährte 24-Nadel-Technologie.

Eine breitgefächerte Gruppe – vom Schüler über den Heimanwender bis hin zum Freiberufler – findet im P 2200 die ideale Drucklösung. Anwendern, die schon seit langem auf der Suche nach einem preisgünstigen Drucker für ihren



12 Schriftartenkassetten zusätzlich erhältlich.

Computer sind, eröffnet er die Möglichkeit, Druckergebnisse in bewährter NEC-Qualität zu erzielen.

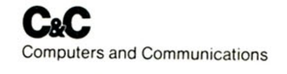
Damit ist der P 2200 die optimale wirtschaftliche und technische Alternative für alle, die sich bei gleichem finanziellen Aufwand bisher nur mit antiquierten 9-Nadel-Druckern begnügen mußten.

Für Beratung und Service stehen Ihnen Ihr NEC Drucker-Fachhändler und sein Team zur Verfügung.



Weitere Informationen erhalten Sie von:

NEC Deutschland GmbH
Klausenburger Straße 4, 8000 München 80
Tel.: 0 89/9 30 06-0, Telefax: 0 89/93 77 76/8
Telex: 5 218 073 und 5 218 074 nec m



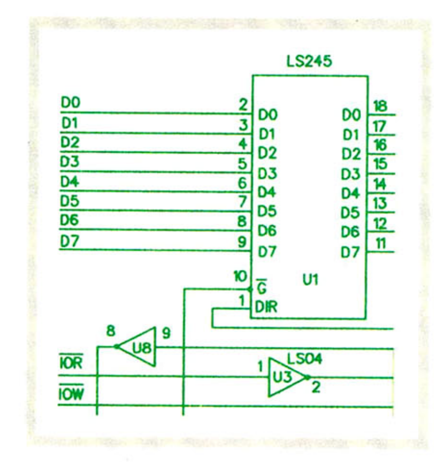
NEC PINWRITER UND TEXTVERARBEITUNG



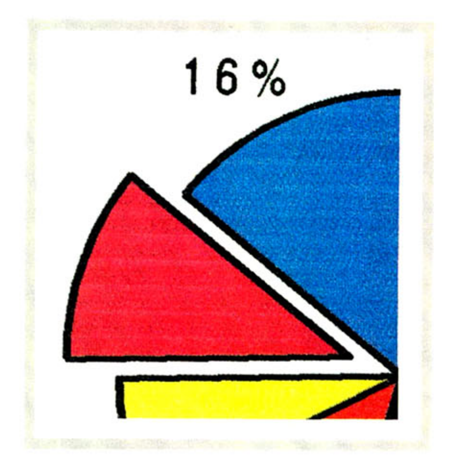


12 Zeichen pro Zoll i
20 Zeichen pro Zoll im Kleindruck -
17 Zeichen pro Zoll im Kleindr
15 Zeichen pro Zoll im Schr
12 Zeichen pro Zoll i
10 Zeichen pro Zol
20 Zeichen pro Zoll in Briefqualitä
17 Zeichen pro Zoll in Briefqu
15 Zeichen pro Zoll in Bri
12 Zeichen pro Zoll i
10 Zeichen pro Zol

Verschiedene Zeichen …



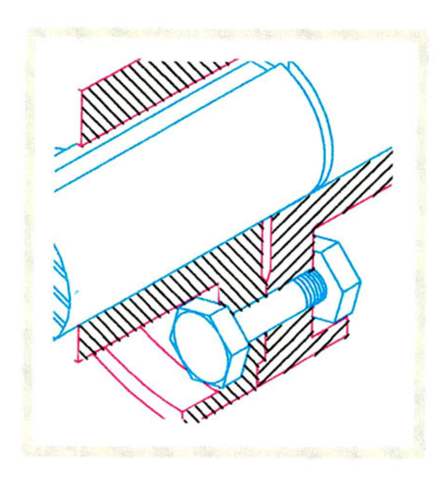
Höchste Druckpräzision, …



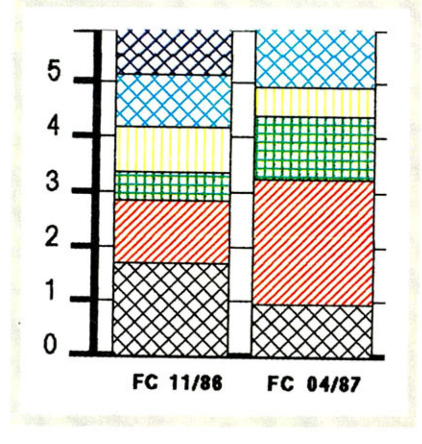
Mit vielen Farben …



… in vielen Kombinationen …



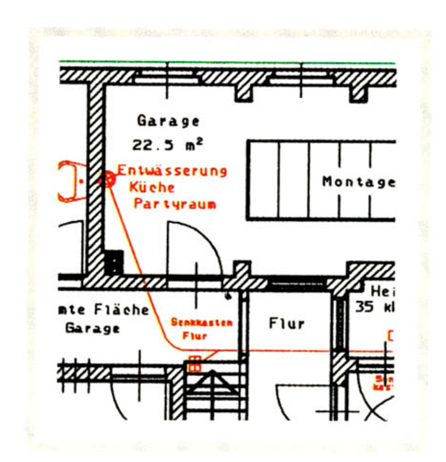
… kombiniert mit Farbe, …



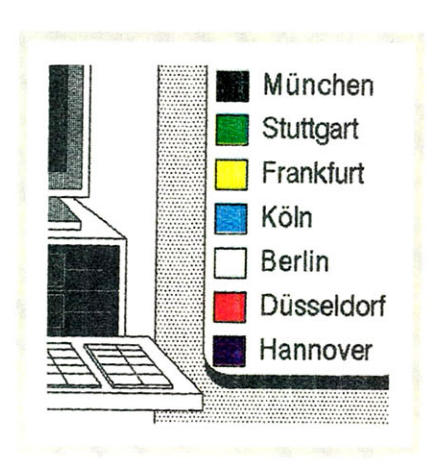
… in allen Schattierungen …



… bilden perfekte Texte.



… liefert korrekte Pläne.



… gelingt jede Präsentation.

Alle Abbildungen sind im Maßstab 1:1 unretuschiert dargestellt und wurden mit NEC Pinwritern ausgedruckt.

NEC PINWRITER UND DESKTOP PUBLISHING

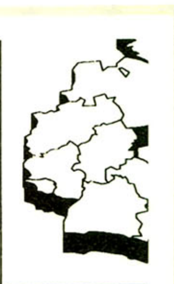
ZEIT
HDENKEN
s auf dieser Welt, das
d ein wenig schlechter

Exakte Schriften . . .



. . . und brillante Grafiken . . .

em Verlag, der
e ich mich für
n.
führung einer breiten
rbekampagnen. Enge
in der Lage, Arbeiten
chungsaufwand und
weise:
erausgegeben
hneten Werke
n Illustrationen



Personen, Organis
in der Bundesrepu
Freiberu

. . . für Ihre Publikationen.

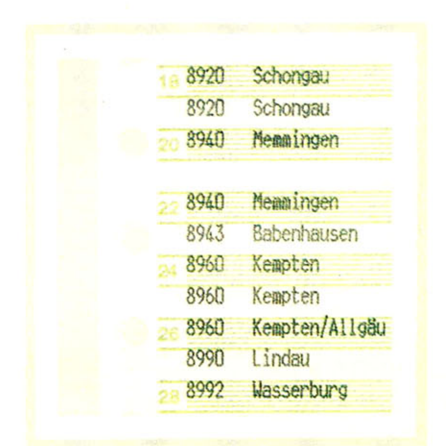
NEC PINWRITER UND LISTENDRUCK



Mit Präzision . . .

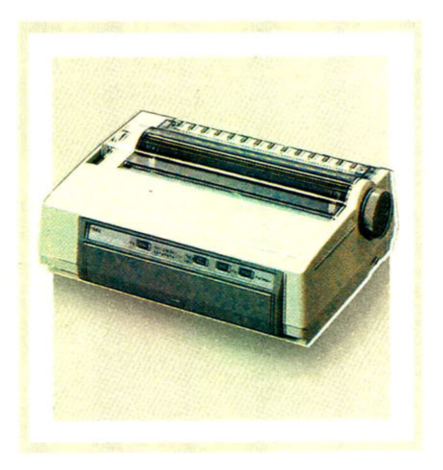


. . . und Geschwindigkeit . . .



. . . große Datenmengen bewältigen.





Pinwriter P2200 (für s/w)



Pinwriter P9 XL (für s/w und color)

Bitte besuchen Sie uns in Halle 6 / Stand H 32/H 46

HANNOVER MESSE CeBIT '88
Welt-Centrum Büro · Information · Telekommunikation
16. – 23. MÄRZ 1988

Weitere Informationen erhalten Sie von Ihrem NEC Drucker-Fachhändler.



NEC Deutschland GmbH
Klausenburger Straße 4, 8000 München 80
Tel.: 0 89/9 30 06-0, Telefax: 0 89/93 77 76/8
Telex: 5 218 073 und 5 218 074 nec m

24-BIT PIO FÜR 15 MARK

Ein Nachteil beim Atari ST ist das Fehlen einer parallelen Ein-/ Ausgabeeinheit (kurz PIO), mit der man steuern oder messen kann.

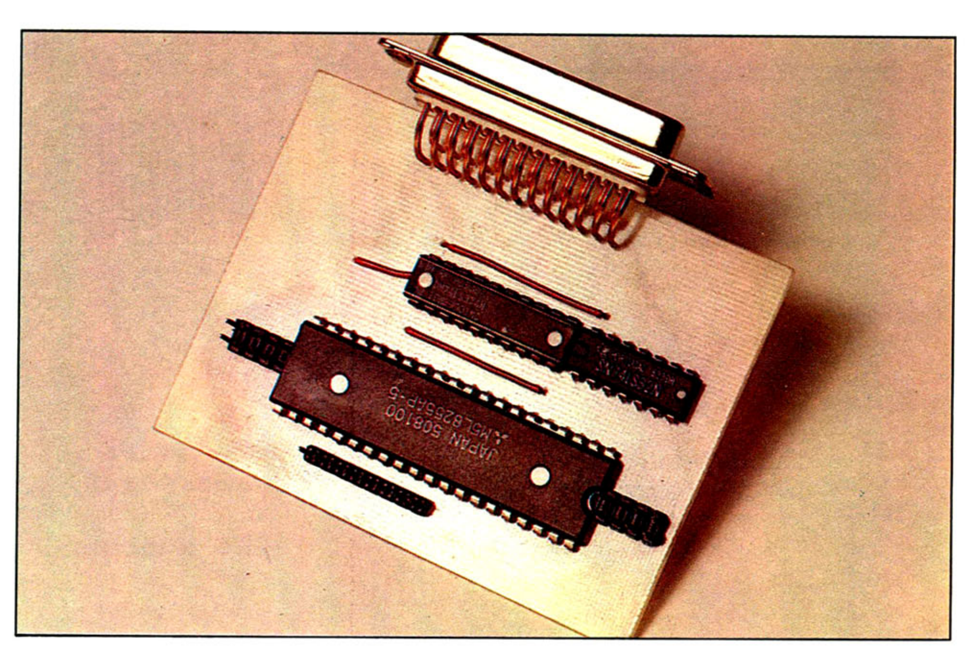
Es gibt bereits diverse Lösungen, die aber allesamt eines gemeinsam haben: sie sind entweder relativ teuer, oder sie blockieren wichtige Anschlüsse (DMA, Rom-Modul, etc.). Auch diese Schaltung blockiert einen Anschluß, nämlich die Centronicsschnittstelle. Aber sie ist billig (ca. 15 DM), und sie ermöglicht es aufgrund des verwendeten Bausteins (eine 8255 PIO von intel), trotzdem einen Drucker anzuschließen.

Das Prinzip der Schaltung

Am Centronicsanschluß befinden sich 10 Leitungen: 8 Datenleitungen und zwei Leitungen, die normalerweise für das Handshaking mit dem Drucker zuständig sind. Die 8255 PIO braucht aber dummerweise 6 Steuerleitungen und 8 Datenleitungen. Nun, die Schaltung benutzt folgenden Trick: Es wird mit den beiden Handshakeleitungen zwischen Daten für die PIO und Steuerdaten, die in einen zusätzlichen 8 Bit-Zwischenspeicher geschrieben werden unterschieden. Da der Baustein weder ein Taktsignal noch ein kompliziertes Timing benötigt, reicht dies aus, um ihn anzusteuern. Ein zweites Problem ist die Stromversorgung, am Centronicsport ist nämlich keinerlei benutzbare Spannung herausgeführt. Aber auch hier gibt es eine einfache Lösung: das Pin 13 des Druckeranschlusses ist nicht belegt. Man kann nun den Atari öffnen und dieses Pin mit +5V verbinden. Die +5V sucht man am besten mit einem Voltmeter in der Nähe des Netzteils.

Eine kurze Beschreibung des 8255

24 programmierbare I/O-Anschlüsse (aufgeteilt in 2 Gruppen zu 8 Bit und 2 Gruppen zu 4 Bit, wobei die zwei 4-Bit-



Prototyp der PIO

Gruppen zu einer 8-Bit-Gruppe zusammengefaßt werden können, um dann drei 8-Bit Gruppen zu bekommen) Möglichkeit, einzelne Bits zu setzen oder zu löschen, Programmierbarkeit des Bausteins in 3 verschiedenen Modi:
- Normale Ein-/Ausgabe (0)
- Impulsgesteuerte Ein-/Ausgabe (1)
- Bidirektionaler Bus (2)

Der Baustein hat 2 Adreßleitungen, mit denen man 4 Register adressieren kann. Die Adressen 0 bis 2 werden für die 3 I/O-Ports verwendet, Adresse 3 wird zur Programmierung der PIO verwendet. Wenn das Bit 7 des in dieses Register geschriebenen Werts 1 ist, so werden die Bits 0-6 als Konfigurationsbefehle verstanden: (Modus 0)

Befehlsnummer: (Hexadezimal)	Port Ø	Port 1	Port 2, unten	Port 2, oben
$8Ø	Ausgang	Ausgang	Ausgang	Ausgang
$81	A	A	Eingang	A
$82	A	E	A	A
$83	A	E	E	A
$88	A	A	A	E
$89	A	A	E	E
$8A	A	E	A	E
$8B	A	E	E	E
$9Ø	E	A	A	A
$91	E	A	E	A
$92	E	E	A	A
$93	E	E	E	A
$98	E	A	A	E
$99	E	A	E	E
$9A	E	E	A	E
$9B	E	E	E	E

Ist das Bit 7 Null, so können einzelne Bits des Ports 3 gesetzt oder gelöscht werden:

Bit 0: 0 bedeutet Bit löschen, 1 = Bit setzen
Bit 1-3: Nummer des Bits von 0 bis 7
Bit 4-6: unbenutzt
Bit 7: muß 0 sein !

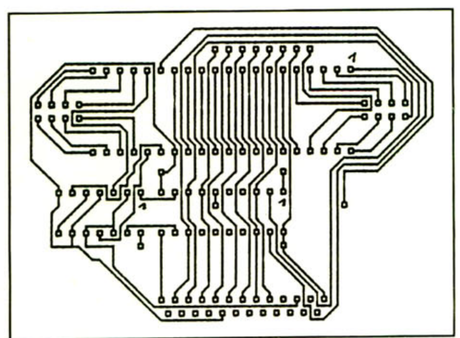
Beispiel: Der Wert $D setzt Bit 6.

Die Modi 1 und 2 interessieren hier nicht weiter, es sei nur erwähnt, daß man z.B. zwei Drucker ansteuern (Modus 1) oder eine IEC-Schnittstelle emulieren könnte (Modus 2). Wer weitere Informationen möchte, sei hier auf das Datenblatt von intel verwiesen (Adresse später).

Schaltplan

Bauteilliste

1 PIO 8255 oder 82C55 (sparsamer)
1 74 LS 32
1 74 LS 374
1 D-Sub Stecker, 25-polig mit abgewinkelten Anschlüssen
Platine (Layout siehe Zeichnung)



Das Steuerregister wird durch einen 74 LS 374 dargestellt, der 8 D-Flipflops beinhaltet. Es ist folgendermaßen aufgebaut:

Bit 0: /RD8255 ; die Leseleitung für die 8255 PIO, wenn auf Low-Pegel, kann gelesen werden.
Bit 1: /WR8255 ; die Schreibleitung, wenn auf Low-Pegel, kann geschrieben werden.
Bit 2: RESET ; die Resetleitung für die 8255, versetzt die PIO in Einschaltzustand, wenn auf High-Pegel

Bit 6: A1 ; das höherwertige Bit der Registernummer
Bit 7: A0 ; das niederwertige Bit der Registernummer

Ein Lesezugriff auf Register 2 läuft dann so ab:
- /RD8255 auf 0, /WR8255 auf 1, RESET auf 0 , A1 auf 1 und A0 auf 0 setzen.
- dieses Steuerwort schreiben
- Centronics als Eingang programmieren
- /CE8255 auf 0 setzen (=PIO anwählen)
- Wert lesen und merken
- /CE8255 auf 1 setzen (=PIO deselektieren)
- Centronics wieder Ausgang

Die Pinbelegung der Stiftleisten ist auf Zeichnung 2 (dem Bestückungsplan) angegeben. Die drei dick eingezeichneten Lötbrücken müssen vor den ICs eingelötet werden ! Bei der Verwendung von Portleitungen als Eingang müssen diese mit einem Pullup-Widerstand versehen werden, d.h. sie werden über einen 10 Kiloohm-Widerstand mit +5V verbunden. Wird dies nicht gemacht, "rauscht" der Eingang. Bei der Verwendung als Ausgang ist dies nicht nötig.

Die Software-einbindung:

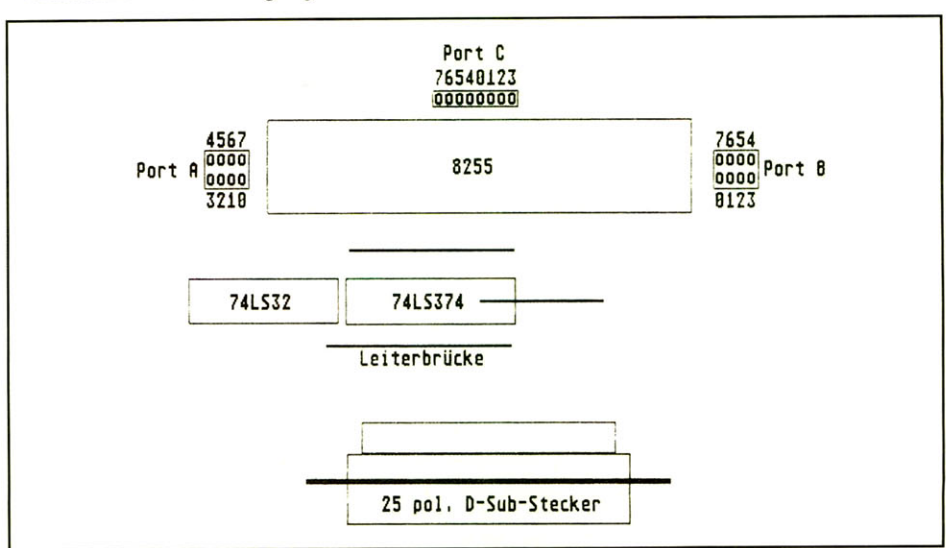
Es existiert Steuersoftware für GFA-BASIC und für das Megamax C. Das Prinzip ist bei beiden Programmen gleich, nur das Einbinden ist in C einfacher, da man hier direkt Assemblermnemonics verwenden kann, während man in BASIC Maschinenprogramme entweder nachladen oder in Stringvariablen halten muß.

Zuerst zum GFA-BASIC:

Die Einbindung ist als Unterprogrammsammlung realisiert, die Sie, wenn Sie sie benötigen, hinzuMERGEn müssen.

Die Unter-programme sind:

Init_all: Initialisiert das MC-Programm und den PIO-Baustein (muß immer vor der ersten Benutzung aufgerufen werden !!!)
Lese_pio(Port%,*Variable%): Liest aus dem PIO-Register Port% in Variable%.



Schreibe_pio(Port%,Wert%):
Schreibt den angegebenen Wert ins PIO-Register Port%.

Init_drucker:
Schaltet die Centronics-schnittstelle wieder auf Druckerbetrieb um.

(Siehe Listing PIO.LST)

Zum Megamax C:

Die Einbindung wurde als .H File realisiert, d.h., wann immer Sie die PIO in C benötigen, schreiben Sie am Programmanfang

#include "pio.h" .

Zum Aufruf der PIO-Routinen wird die Funktion 'pio_access' verwendet. Als Argument benötigt man den Befehl, die Registernummer und den Wert. Die Funktion liefert bei Leseoperationen einen int Wert zurück.

Initialisieren: pio_access(_INIT,0,0); schaltet die PIO ein, muß immer vor erster Benutzung aufgerufen werden.

Lesen: x=pio_access(_LESE,2,0); liest aus Port 2 in Variable x

Schreiben: pio_access(_SCHREIBE, 0,0xAA); schreibt den Wert $AA in Port 0

Druckerbetrieb: pio_access(_ENDE,0,0); schaltet auf Druckerbetrieb um.

(Siehe Listing 2+3)

Jörg Falkenberg
Jürgen Landler

Literatur:

intel Microsystem Components Hand book Volume II, Seite 5-273 (Adresse: intel, Seidlstraße 27, 8000 München 2)

ATARI ST Intern, Data Becker

PIO.LST

```
 '
 ' Treibersoftware für 24-Bit PIO mit 8255
 ' Software (c)1987 Jürgen Landler
 ' Hardware (c)1987 Jörg Falkenberg
 '
 @Init_all
 Print Hex$(Bef%)
 @Schreibe_pio(3,&H89)
 Repeat
   @Lese_pio(2,*I%)
   Print At(1,2);I%'
   @Schreibe_pio(0,0)
   @Schreibe_pio(1,0)
   Pause 5
   @Schreibe_pio(0,255)
   @Schreibe_pio(1,255)
   Pause 5
 Until Inkey$="s"
 @Init_drucker
End
'
' Liest MC-Unterprogramme ein
'
Procedure Init_all
  Local A$,I%
  Restore Pio_code
  Bef$=""
  Do
    Read A$
    Exit If A$="-1"
    Bef$=Bef$+Mki$(Val("&H"+A$))
  Loop
  Bef%=Varptr(Bef$)
  @Init_pio                    ! Druckerport initialisieren
Return
'
' Schaltet den Druckerport auf Ausgang, und 8255 auf
   Standby
'
Procedure Init_pio
  Void Xbios(38,L:(Bef%+&HA0))
Return
'
' schaltet Centronics wieder auf Druckerbetrieb um
'
Procedure Init_drucker
  Void Xbios(38,L:(Bef%+&H60))
Return
'
' schreibt Wert_ in Register Reg_
'
Procedure Schreibe_pio(Reg_%,Wert_%)
  Dpoke Bef%,Wert_%
  Dpoke Bef%+2,Reg_%
  Void Xbios(38,L:(Bef%+&HEA))
Return
'
' liest aus Reg_ in durch Wohin_ adressierte Variable
'
Procedure Lese_pio(Reg_%,Wohin_%)
  Dpoke Bef%+2,Reg_%
  Void Xbios(38,L:(Bef%+4))
  *Wohin_%=Dpeek(Bef%)
Return
'
' MC-Unterprogramme
'
Pio_code:
Data 0000,0000,40C2,46FC,2700,08F9,0000,00FF
Data FA05,41F9,00FF,8800,323A,FFE8,7008,E219
Data E211,E210,E511,E210,6100,00A0,10BC,0007
Data 1010,0240,007F,1140,0002,10BC,000F,08B9
Data 0000,00FF,FA01,1039,00FF,8800,08F9,0000
Data 00FF,FA01,43FA,FFAA,3280,6134,46C2,4E75
Data 40C2,343C,2700,41F9,00FF,8800,08B9,0000
Data 00FF,FA05,08B9,0000,00FF,FA01,6112,10BC
Data 000E,1010,0000,0020,1140,0002,46C2,4E75
Data 10BC,0007,1010,0000,0080,1140,0002,4E75
Data 40C2,46FC,2700,41F9,00FF,8800,08F9,0000
Data 00FF,FA05,08F9,0000,00FF,FA01,61D2,70FF
Data 6108,70C3,6104,46C2,4E75,10BC,000F,1140
Data 0002,10BC,000E,1010,0000,0020,1140,0002
Data 0200,00DF,1140,0002,4E75,40C2,46FC,2700
Data 08F9,0000,00FF,FA05,41F9,00FF,8800,323A
Data FF02,7004,E219,E211,E210,E511,E210,61BA
Data 10BC,000F,117A,FEEB,0002,08B9,0000,00FF
Data FA01,08F9,0000,00FF,FA01,46C2,4E75
Data -1
```

Weitere Listings nächste Seite

PIO.H

```
#define _LESE 1
#define _SCHREIBE 2
#define _INIT 3
#define _ENDE 4

int pio_access(befehl,port,wert)
int befehl,port,wert;
{
   int zurueck;
   long oldstack;
   oldstack=Super(0L);
   switch(befehl)
   {
   case _LESE:
      asm
      {
         move.w SR,D2
         move.w #0x2700,SR         ; Interrupts aus
         bset #0,0xfffa05
         lea 0xff8800,A0           ; Basisadresse des
                                     Soundchips (AY)
         move.w port(A6),D1        ; Register, Read
         moveq #8,D0               ; /WR8255 auf High,
                                     /RD8255 auf Low
         ror.b #1,D1               ; Portadresse in Bit
         roxr.b #1,D1              ; 6 und 7 schieben
         roxr.b #1,D0
         roxl.b #2,D1
         roxr.b #1,D0
         bsr stout                 ; Steuerwort schreiben
         move.b #7,(A0)            ; AY Eingang
         move.b (A0),D0            ; Register 15 als
                                     Eingang
         andi.w #127,D0            ; programmieren
         move.b D0,2(A0)
         move.b #15,(A0)           ; Daten lesen
         bclr #0,0xfffa01
         move.b 0xff8800,zurueck(A6)
         bset #0,0xfffa01
         bsr ayaus                 ; Reg. 15 wieder
                                     Ausgang
         move.w D2,SR              ; Interrupts wieder an
         bra lese_ende

      stout:
         move.b #15,(A0)           ; Daten in AY-Register
         move.b D0,2(A0)           ; 15 schreiben
         move.b #14,(A0)           ; und mit Strobe-
                                     Impuls
         move.b (A0),D0            ; in Steuerregister
                                     (LS 374)
         ori.b #0x20,D0            ; übertragen
         move.b D0,2(A0)
         andi.b #0xdf,D0
         move.b D0,2(A0)
         rts

      ayaus:
         move.b #7,(A0)            ; Port 15 auf
         move.b (A0),D0            ; Ausgang umschalten
         ori.b #128,D0
         move.b D0,2(A0)
         rts

      lese_ende:
      }
   break;

   case _SCHREIBE:
      asm
      {
         move.w SR,D2
         move.w #0x2700,SR
         bset #0,0xfffa05
         lea 0xff8800,A0
         move.w port(A6),D1        ; Port Nummer
         moveq #4,D0               ; /WR8255=Low,
                                     /RD8255=High
         ror.b #1,D1               ; siehe oben
         roxr.b #1,D1
         roxr.b #1,D0
         roxl.b #2,D1
         roxr.b #1,D0
         bsr stout
         move.b #15,(A0)           ; Daten in Register
         move.b wert+1(A6),2(A0)   ; 15 schreiben
         bclr #0,0xfffa01          ; /CE8255 auf low...
         bset #0,0xfffa01          ; ...und wieder high
         move.w D2,SR
      }
      break;

   case _ENDE:
      asm
      {
         move.w SR,D2
         move.w #0x2700,D2
         lea 0xff8800,A0
         bclr #0,0xfffa05          ; Busy wieder Eingang
         bclr #0,0xfffa01
         bsr ayaus
         move.b #14,(A0)           ; Strobe auf High
         move.b (A0),D0
         ori.b #0x20,D0
         move.b D0,2(A0)
         move.w D2,SR
      }
      break;

   case _INIT:
      asm
      {
         move.w SR,D2
         move.w #0x2700,SR
         lea 0xff8800,A0
         bset #0,0xfffa05          ; Busy jetzt Ausgang
         bset #0,0xfffa01          ; und gleich High
                                     setzen
         bsr ayaus
         moveq #0xff,D0            ; 8255 Reset geben
         bsr stout
         moveq #-0x3D,D0           ; RESET wieder low
         bsr stout
         move.w D2,SR
      }
      break;
   }
   Super(oldstack);
   return(zurueck);
}
```

PIO.C

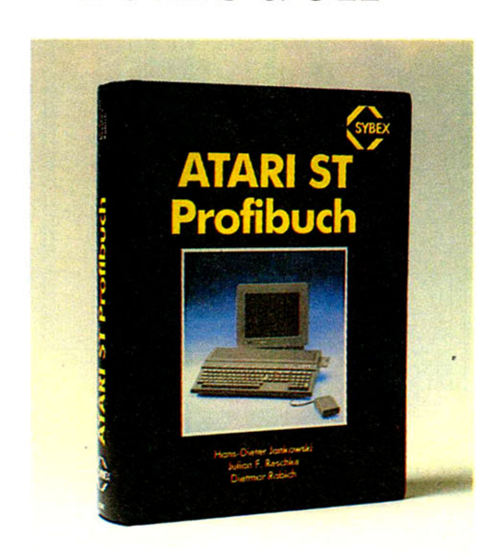
```
/*********************************************
 *  Treibersoftware für 24-Bit PIO mit 8255   *
 *    Software (c) 1987 Jürgen Landler        *
 *    Hardware (c) 1987 Jörg Falkenberg       *
 *********************************************/

#include <osbind.h>
#include "pio.h"

main()
{
   int a=1;
   int b=255;
   int c=0xaa;
   int i;

   puts("\033E Ausgabetest:\n\n");
   pio_access(_INIT,0,0);
/* zuerst Reset */
   pio_access(_SCHREIBE,3,0x80);
/* alle Ports Ausgang */
   do
   {
     pio_access(_SCHREIBE,0,a);
     a=(a<<1); a=( (a&0x100) >> 8 | a ) &0xff;
   /* Bit um eins nach links rotieren */
     pio_access(_SCHREIBE,1,b);
     b^=0xff;
   /* alle Bits in Port B invertieren */
     pio_access(_SCHREIBE,2,c);
     c^=0xff;
   /* auch hier alle Bits invertieren */
     for(i=0;i++<35;Vsync());
   /* 1/2 Sekunde warten */
   } while(!Cconis()); /* bis Taste gedrückt wird */
   pio_access(_ENDE,0,0);
}
```

Das ATARI ST-Profibuch



Schon vor einigen Monaten hielt ich zwei Bücher mit den Titeln ARBEITEN mit GEM (Band 1, die AES-Bibliothek und Band 2, die VDI-Bibliothek) in der Hand, um sie zu beurteilen. Damals befand ich diese beiden Bücher für durchaus brauchbar und arbeite auch heute noch damit. Allerdings behandeln sie ausschließlich den grafischen Teil des ATARI ST, das GEM. Die Betriebssystemteile GEMDOS, XBIOS, BIOS bleiben genauso unerwähnt wie die Hardware unseres liebgewonnenen Rechners. Oft wird die Frage gestellt, welches Buch man sich kaufen soll, um mit dem ATARI ST arbeiten zu können: Es fällt auf, daß es praktisch kein Buch auf dem Markt gibt, das alle Bereiche des ST bearbeitet, so daß man gleich mehrere Bücher empfehlen muß - ein teures und mühsames Unterfangen, die richtigen Bücher zu kaufen.

Jetzt hat die Firma SYBEX unter der Vielzahl ihrer Neuerscheinungen ein Buch mit dem Namen "ATARI ST-Profibuch" auf den Markt gebracht, das kurz betrachtet werden soll. Ein Blick in das Inhaltsverzeichnis zeigt, daß die Autoren sich viel vorgenommen haben: Die Software wie das BIOS, XBIOS, GEMDOS, VDI und AES und die Hardware wie die CPU, das Grafiksystem, den Soundgenerator, den MFP, die Schnittstellen (parallel, seriell, Floppy, DMA) und vieles mehr - eine Menge an Information, die schwer in einem Buch unterzubringen ist, selbst wenn dieses über 730 (!) Seiten hat. Das Buch ist, wie in der obigen Aufzählung zu erkennen, in die verschiedenen Teile des ATARI ST untergliedert. Die Software-Schnittstellen, also die Befehle und Funktionen des Betriebssystems, sind in C dargestellt, der meistgebrauchten Programmiersprache in der Softwareentwicklung auf dem ST. Schön wäre es sicherlich gewesen, für jede Routine eine eigene Seite zu nehmen, wahrscheinlich hätte das Buch dann aber die Ausmaße eines Telefonbuchs der Stadt Berlin. Die wenig bekannten Routinen wie zum Beispiel fom_keybd und form_button, wie auch eine Beschreibung des Blitters und seiner Befehle sind auch vorhanden, leider ist die Beschreibung der beiden form-Befehle nicht sehr ausführlich. Auch die Hardwarebeschreibung kann man sicherlich als gelungen und ausführlich - es lebe das Schema und das Bild - bezeichnen, allerdings fehlen mal wieder einige Beispiele, wie man die erklärten Befehle - beispielsweise des Tastaturprozessors oder des FDCs - in der Praxis benutzt - schade.

Ein Lob dem Stichwortverzeichnis

Am Ende des Buches befindet sich ein Anhang von circa 80 Seiten, in dem Tabellen, Strukturen, Ablaufpläne, Pinbelegungen und vieles mehr zu finden sind. Den krönenden Abschluß bildet das Inhaltsverzeichnis. Viele wirklich gute Bücher schneiden bei der praktischen Benutzung äußerst schlecht ab, da man bestimmte Themen erst nach langem Suchen im Buch findet. Da bei guten Textverarbeitungen das Anlegen eines Stichwortverzeichnisses kein Problem mehr ist, stimmt diese Tatsache immer wieder nachdenklich. Hier möchte ich dem vorliegenden Buch ein großes Lob aussprechen ! Das fast 30 (dreißig) Seiten umfassende Stichwortverzeichnis läßt den Leser in kaum einem Fall im Stich und wertet das Buch gegenüber anderen Büchern stark auf.

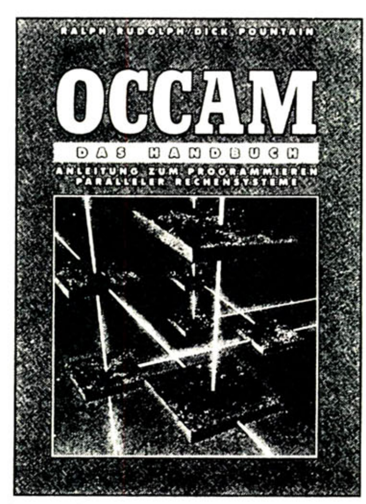
Resümee

Die oben genannten negativen Punkte sind in keinster Weise gravierend und sollten nur als Anregung zur nächsten Auflage dienen; auch wenn das Buch nicht optimal ist, sind diese Punkte auf keinen Fall ein Grund, es nicht zu kaufen, denn die Themen sind gut dargestellt. Obwohl dieses Buch den etwas prahlerischen Namen PROFIBUCH hat muß man feststellen, daß ein Arbeiten sehr einfach und informativ ist, allerdings ist es eher ein Nachschlagewerk ("Profibuch") als ein Lernbuch, aber beides in einem ist wohl kaum umzusetzen. Sollte mich jemand fragen, welches Buch ich ihm empfehle, werde ich dieses Buch nennen, denn ein Buch, das so umfassend ist und trotzdem nur 58,- DM kostet, sucht seinesgleichen...

Stefan Höhn

Hans-Dieter Jankowski, Julian F. Reschke, Dietmar Rabich
ATARI ST PROFIBUCH
Düsseldorf 1987
SYBEX-Verlag GmbH, 760 Seiten
DM 58,-
ISBN 3-88745-501-0

OCCAM - Das Handbuch



Nach Ataris Ankündigung, ein Transputersystem auf den Markt zu bringen (s. ST-Computer 1/88) und der Vorstellung von Kumas KMAX (s. ST-Computer 9/87) wird für Atari-Fans auch die Sprache der Transputer interessant: OCCAM. Der Entwickler der Transputer-Chips, INMOS hat inzwischen eine rege Veröffentlichungstätigkeit entwickelt, unter anderem die "Tutorial Introduction" zu OCCAM, die hier als deutschsprachige Übersetzung vorliegt.

Das Buch gibt eine Einführung in die OCCAM-Konzepte, die Prozesse und die sie verbindenden Kanäle. Das erste Kapitel beschreibt etwas allgemeiner die grundsätzlichen Überlegungen zu Parallelität, Synchronisation und Kommunikation, und wie sie in OCCAM wiederzufinden sind. Im zweiten Kapitel geht es dann zur Sache: von den primitiven Prozessen über die zentralen Sprachkonstrukte SEQ, PAR und ALT zu Typende-

finitionen, Prozeduren und den Kommunikationsstatements in OCCAM. Hier findet sich das wirklich neue an OCCAM, die programmiersprachliche Darstellung von Parallelität.
Der dritte Abschnitt beschreibt die arithmetischen Operatoren, Felder, getypte Kanäle und String- und Zeichenbehandlung. Daran schließen sich die fortgeschritteneren Möglichkeiten von OCCAM an, wie replizierte Prozesse, Timer-Typen, die Konfiguration eines Programms auf die reale Hardware und das Retyping, also die Veränderung eines Variablentyps während der Laufzeit. Das Kapitel "Die Praxis" geht auf allgemeine Probleme zu Programmtermination und -entwurf ein.
Abschließend wird die Syntaxdefinition aufgelistet und eine gut kommentierte Literaturliste geboten. Ein Register rundet das Buch ab. Der Text ist leicht lesbar und von vielen Beispielprogrammen (oder besser Prozessen) begleitet, wobei immer wieder auf Unterschiede zu bekannten Programmiersprachen hingewiesen wird. Das Buch ist mit der typisch anglophilen Lockerheit geschrieben, was nicht verwundert, wenn man weiß, daß Dick Pountain freier Mitarbeiter der führenden amerikanischen Computerzeitschrift BYTE ist.
Eigentlich fehlt ein kleines Kapitel über die grundsätzlichen Probleme der parallelen Programmierung. Begriffe wie Deadlock, Starvation oder theoretische Überlegungen zur Kommunikation von Prozessen sind dabei unabdingbar. Allerdings mögen diese Themen für Anfänger harte Brocken sein und füllen selber schon ganze Bücher.
Insgesamt ein interessantes Buch über die Grundlagen der Softwareseite der neuen Transputer-Technik, die in den nächsten Jahren auch für den Hobbyisten erschwinglich werden wird.

Robert Tolksdorf

Dick Pountain / Ralph Rudolph
OCCAM - Das Handbuch
Hannover 1987
Heinz Heise Verlag
127 Seiten
DM 34,80
ISBN 3-88229-001-3

ENDE

Das Tuning-Programm für Ihren Atari ST

Erst prüfen, dann kaufen
Schauen Sie sich dieses Werk in Ruhe an: 10 Tage lang dürfen Sie Ihr Ansichtsexemplar unverbindlich zu Hause prüfen.



Interessiert an 68000er-Programmierung in Assembler und C?
Blättern Sie bitte um!

Dieses neue Nachschlagewerk bietet Ihnen

■ **ausführliche Beschreibungen des äußeren und internen Aufbaus der Atari ST-Reihe:** Sie lernen die einzelnen Bausteine wie 68000er-Prozessor, DMA-Controller oder Glue-Baustein im Detail kennen und erfahren, wie diese Komponenten zusammenarbeiten;

■ **das Know-how zur Systemprogrammierung:** Anhand von Beispielen werden Sie mit GEM und den TOS-Komponenten GEM-DOS, BIOS und XBIOS vertraut. Sie lernen Systemroutinen, z. B. für die Mausprogrammierung oder Fenstermanagement, zu nutzen. Den detaillierten Betriebssystemübersichten entnehmen Sie u. a. die Speicheradressen der verschiedenen TOS-Versionen oder den Befehlsvorrat an GEM-, AES- und GEM-VDI-Routinen;

■ **Kurse für erfolgreiches Softwareengineering:** Mit Beispielen aus zentralen Anwendungen wie Grafik oder Dateiverwaltung verfügen Sie gleichzeitig über sofort einsetzbare Lösungen;

■ **Programmiersprachkurse** für C, 68000er-Assembler, GFA-Basic;

■ **Systemtuning durch Hardwareerweiterungen und Bausteinprogrammierung:** Präzise Anleitungen zeigen Ihnen detailliert, wie Sie bei Ihrem Atari eine RAM-Erweiterung oder eine ROM-TOS-Aufrüstung vornehmen. Bauanleitungen mit Platinenlayouts auf Folie, zusammen mit der erforderlichen Betriebssoftware, ermöglichen Ihnen den kostengünstigen Selbstbau eines hochleistungsfähigen Festplattenlaufwerks, eines EPROMers sowie eines universellen Erweiterungsports.

■ **praxiserprobte Musterlösungen** für **Technik/Mathematik** (Logikentwicklungssimulator zur Entwicklung von digitalen Schaltungen, Fouriertransformationen, Matrizenmultiplikation u. a.), **Planung** (z. B. Netzplantechnik), **Grafik** (z. B. Berechnung von Zentral- und Parallelprojektionen, 3D-Grafik, Bilder in 512 Farben bei gleichzeitigem Maus- und Tastaturhandling), **Sound** (Sound-Sampler u. a.) **Add-On-Programme zur Standardsoftware, Desktop-Publishing;**

■ **Tips, Tricks und Utilities:** Sie erhalten **Hardwaretips** (z. B. für den Einsatz eines 5¼"-Laufwerks), **Floppyroutinen** u. a. Datenübernahme von MS-DOS-Disketten), **Druckerutilities** (Ausdruck des Directory), **Systemutilities** (RAM-Disk, EPROM-Programmierung, Autoboot mit integriertem Monitorschutz), **Konvertierutilities** (zum problemlosen Datenaustausch zwischen Atari ST und Amiga); **Komprimierroutinen für Programme;**

■ **den preisgünstigen Diskettenservice**

■ **regelmäßige Ergänzungsausgaben zum Grundwerk** mit aktuellen Informationen rund um den Atari wie z. B. neuentwickelte Hard- und Software, neue Betriebssystemversionen.

Fordern Sie noch heute mit nebenstehender Bestellkarte an:

Erfolgreiches Hard- und Softwaretuning der Atari ST 260/520/1040 und Weiterentwicklungen

Stabiler Ringbuchordner, Format DIN A 4, Grundwerk ca. 400 Seiten, Bestell-Nr. 3700, Preis: DM 92,– (erscheint ca. 1. Quartal 1988)

Alle 2–3 Monate erhalten Sie Ergänzungsausgaben zum Grundwerk mit je ca. 120 Seiten zum Seitenpreis von 38 Pfennig (Abbestellung jederzeit möglich).

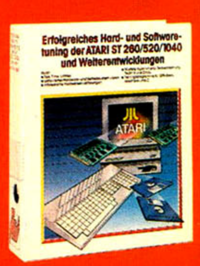


INTEREST-VERLAG
Fachverlag für anspruchsvolle Freizeitgestaltung
Industriestraße 21
D-8901 Kissing
Tel. 08233/23900

Bitte abtrennen

⊘ Ja, senden Sie mir bitte sofort

☐ Expl. **Erfolgreiches Hard- und Softwaretuning der Atari ST 260/520/1040 und Weiterentwicklungen**
Stabiler Ringbuchordner, Format DIN A 4, Grundwerk ca. 400 Seiten, Bestell-Nr. 3700, Preis: DM 92,– (erscheint ca. 1. Quartal 1988)

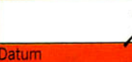


☐ Expl. **„Erfolgreiches Programmieren von 68000er-Systemen in Assembler und C"**
stabiler Ringbuchordner, Format DIN A 4, Grundwerk ca. 400 Seiten, Bestell-Nr. 3400, Preis: DM 92,– (erscheint ca. 1. Quartal 1988)



Zu jedem dieser Werke erhalten Sie alle 2–3 Monate Ergänzungsausgaben zum Grundwerk mit je ca. 120 Seiten zum Seitenpreis von 38 Pfennig (Abbestellung jederzeit möglich).

Unterschreiben Sie hier bitte Ihre Bestellung!
Bei Minderjährigen ist die Unterschrift eines gesetzlichen Vertreters erforderlich. Ohne Ihre Unterschriften kann die Ansichtsbestellung nicht bearbeitet werden.



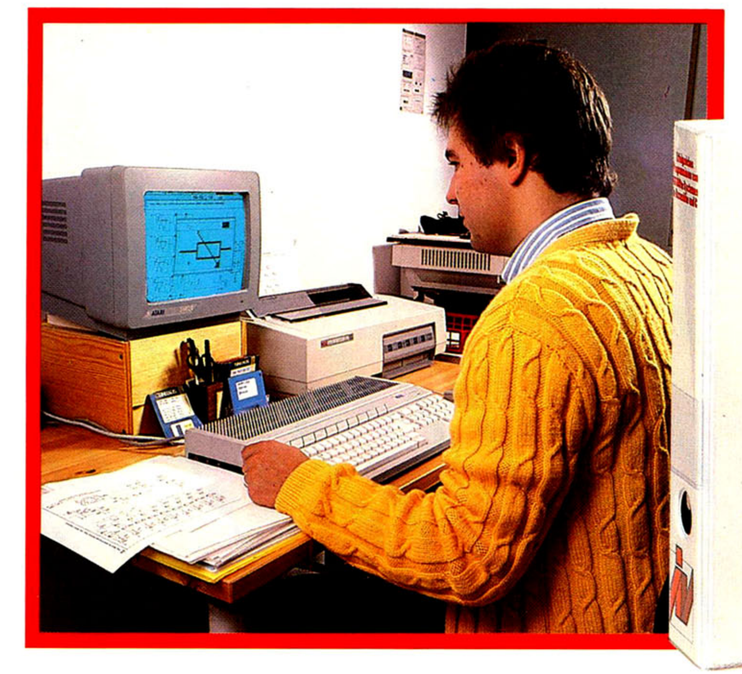
Datum Unterschrift

Bitte unterschreiben Sie auch Ihre Sicherheitsgarantie!
mit der Sie folgendes zur Kenntnis nehmen: Sie haben das Recht, Ihr angefordertes Werk innerhalb von 10 Tagen ab Lieferung an die Bestelladresse zurückzusenden, wobei für die Fristwahrung das Datum der Absendung genügt. Sie kommen dadurch von allen Verpflichtungen aus der Bestellung frei.



Datum Unterschrift

68000er-Systeme perfekt programmieren in Assembler und C



Erst prüfen, dann kaufen
Schauen Sie sich dieses Werk in Ruhe an: 10 Tage lang dürfen Sie Ihr Ansichtsexemplar unverbindlich zu Hause prüfen.

Nutzen Sie die Stärken Ihres Atari!
Jetzt hilft Ihnen dieses neue Nachschlagewerk,
■ effiziente Problemlösungen auf **Betriebssystemebene** zu realisieren
■ anspruchsvolle **Anwenderprogramme** zu entwickeln
■ mit ausgefeilter **Bausteinprogrammierung** eine neue Atari-Dimension zu erschließen.

Die wichtigsten Themen auf einen Blick:
■ **detaillierte Hardware-Beschreibungen** der Prozessorfamilie 68000 und ihrer 8-, 16-, 32-Bit-Peripheriebausteine mit technischen Kennwerten, Pinbelegung, internen Architekturen, Befehlsbeschreibungen;
■ **Betriebssysteme:** Sie lernen die Strukturen der 68000er-Betriebssysteme (wie GEM-TOS. OS-9 und Unix) kennen und erfahren, wie Sie die Systemroutinen zur Optimierung Ihrer Assembler- und C-Programme nutzen;
■ **Softwareengineering:** die optimale Vorgehensweise von der Problemanalyse über die Codierung bis hin zur abschließenden Dokumentation;
■ **Programmierkurse für effiziente Anwender-, System- und Bausteinprogrammierung in Assembler und C.** Damit verfügen Sie zugleich über sofort einsetzbare Routinen, z. B. zur Drucker- und Bildschirmansteuerung;
■ **die C-Compiler-Werkstatt:** Schritt für Schritt programmieren Sie selbst einen C-Compiler inkl. Funktionsbibliotheken;
■ **Assembler-Makrobibliotheken, Anwender- und Funktionsbibliotheken in C;**
■ **Tips und Utilities** wie Schnittstellen- und Backup-Routinen, Fensterverwaltung;
■ **bewährte Musterlösungen** wie zu Rechnerkopplung und Multitasking;
■ **Anleitung für raffiniertes System-tuning,** z. B. durch Einsatz schnellerer Peripheriebausteine/Höhertakten der CPU;
■ **Praxiswissen zu speziellen Einsatzbereichen** wie Messen, Steuern, Regeln;

Dieses Werk veraltet nie
Unsere 68000er-Fachredaktion versorgt Sie regelmäßig und zuverlässig mit aktuellen Informationen über Hardware- und Betriebssystem-Weiterentwicklungen, weiteren Utilities und Musterlösungen sowie Bibliothekserweiterungen in Assembler und C.

Bitte abtrennen

Meine Anschrift:

Name

Vorname

Straße, Haus-Nr.

PLZ/Ort

Nutzen Sie unseren kostenlosen Informations-Service:

☐ Grafik und CAD
☐ Desktop-Publishing
☐ dBase-Programmierung
☐ Messen, Steuern, Regeln mit dem Atari ST
☐ Aktuelles IC-Datenbuch

IN 160043

60 Pfennig die sich lohnen!

Postkarte/Antwort

INTEREST-VERLAG
Fachverlag für anspruchsvolle Freizeitgestaltung
Herrn Michel
Industriestraße 21

D-8901 Kissing

Fordern Sie noch heute mit nebenstehender Bestellkarte an:

Erfolgreiches Programmieren von 68000er-Systemen in Assembler und C

stabiler Ringbuchordner, Format DIN A 4, Grundwerk ca. 400 Seiten. Bestell-Nr. 3400, Preis: DM 92,– (erscheint ca. 1. Quartal 1988)

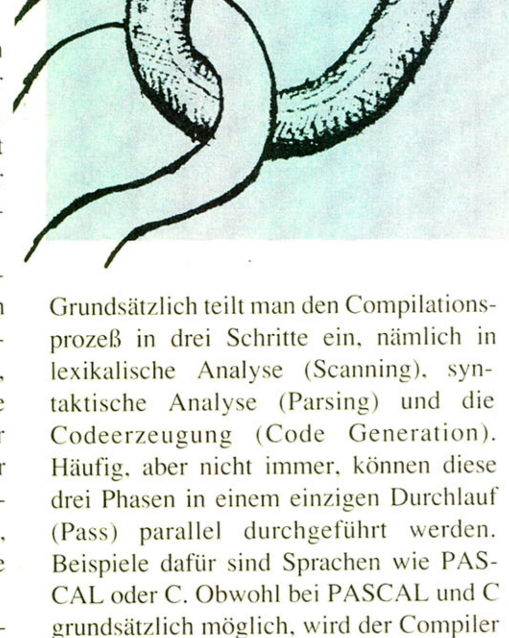
Alle 2–3 Monate erhalten Sie Ergänzungsausgaben zum Grundwerk mit je ca. 120 Seiten zum Seitenpreis von 38 Pfennig (Abbestellung jederzeit möglich).

Industriestraße 21
D-8901 Kissing
Tel. 0 82 33/2 39 00

INTEREST-VERLAG
Fachverlag für anspruchsvolle Freizeitgestaltung

VOM QUELLTEXT ZUM PROGRAMM

NACH EINER KURZEN ERLÄUTERUNG DER FUNKTIONSWEISE EINES INTERPRETERS UND COMPILERS SOLL SCHWERPUNKTMÄSSIG AUF DEN LINKER EINGEGANGEN WERDEN, DER LETZTENDLICH EIN ABLAUFFÄHIGES PROGRAMM ERZEUGT. EIN SOLCHER LINKER FÜR DAS GST-FORMAT, DER MIT DEM LATTICE C-COMPILER ENTSTAND, LIEGT ALS LISTING VOR.



Sicherlich weiß jeder, der in höheren Programmiersprachen programmiert, darüber Bescheid, daß der Computer diese Sprache nicht so ohne weiteres versteht. Er akzeptiert nämlich nur Anweisungen in seiner Maschinensprache. Auch allgemein bekannt dürfte sein, daß dies beim ATARI ST die Maschinensprache des 68000-Prozessors ist. Solche Maschinensprachen sind für den Menschen beinahe unlesbar, denn die Maschinenbefehle sind allesamt durch Zahlen codiert, deren Bedeutung man in einer Tabelle nachschlagen kann. Dies kann man auch einem Assembler überlassen, dadurch wird die Maschinensprache lesbarer und leichter anwendbar. Der Assembler übersetzt zum Beispiel Texte wie 'ADD #1,D0' in den codierten Maschinenbefehl $5240.

Aber auch in einer solchen Assemblersprache ist das Programmieren noch recht mühselig. Man stelle sich etwa die Auswertung eines Ausdrucks, wie zum Beispiel z=3*(x+y), vor. In der 68000-Assemblersprache hätte man dazu etwa folgendes zu schreiben:

```
MOVE   x,D0
ADD    y,D0
MULS   #3,D0
MOVE   D0,z
```

Die Schreibweise z=3*(x+y) ist eben doch um einiges verständlicher. Deshalb existieren verschiedene höhere Programmiersprachen, die es ermöglichen, Programme in bedeutend lesbarerer Form zu schreiben. Allerdings müssen diese Programme, bevor sie ausgeführt werden, in die für den Computer verständliche Form gebracht werden. Dafür gibt es grundsätzlich zwei Möglichkeiten, nämlich Interpreter und Compiler.

Der Interpreter

Bekanntestes Beispiel dafür ist sicherlich der BASIC-Interpreter. Ein Interpreter durchläuft den Programm-Quelltext, analysiert dabei Befehl für Befehl und führt entsprechende Aktionen durch. Bei einer Zuweisung 'neu=alt' passiert etwa folgendes:

Das Wort 'neu' wird Buchstabe für Buchstabe gelesen, so lange, bis das '=' Zeichen erreicht ist. Dann wird in einer Variablentabelle, die der Interpreter anlegt, nach der Variablen 'neu' gesucht. Ist diese gefunden, wird das Wort 'alt', wieder Buchstabe für Buchstabe gelesen und in der Variablentabelle nach 'alt' gesucht. Nachdem auch diese Variable gefunden wurde, kann der Wert von 'alt' auf die Variable 'neu' kopiert werden.

Wie man sieht, verlangt selbst eine einfache Zuweisung eine Menge Arbeit vom Interpreter. Stellt man sich nun noch vor, daß diese Zuweisung innerhalb einer Schleife stattfindet, kann man erahnen, warum Interpreter oft so langsam sind.

Der Compiler

...geht einen anderen Weg. Er analysiert den Quelltext nur einmal und übersetzt ihn vollständig in ein Maschinenprogramm. Dabei wird jeder Variablen eine Speicheradresse zugeordnet, und die Zuweisung 'neu=alt' wird beispielweise in den Maschinenbefehl 'MOVE 1000,1002' übersetzt, wobei 1000 der Speicherplatz für die Variable 'alt' und 1002 der Speicherplatz für die Variable 'neu' ist. Man kann sich jetzt leicht vorstellen, wieviel schneller das Programm wird, wenn es compiliert ist.

Grundsätzlich teilt man den Compilationsprozeß in drei Schritte ein, nämlich in lexikalische Analyse (Scanning), syntaktische Analyse (Parsing) und die Codeerzeugung (Code Generation). Häufig, aber nicht immer, können diese drei Phasen in einem einzigen Durchlauf (Pass) parallel durchgeführt werden. Beispiele dafür sind Sprachen wie PASCAL oder C. Obwohl bei PASCAL und C grundsätzlich möglich, wird der Compiler trotzdem oft in mehrere Phasen eingeteilt, um Speicher zu sparen. Die Kommunikation der Phasen untereinander erfolgt dann durch Zwischenfiles. Doch nun zur eigentlichen Arbeit des Compilers:

In der lexikalischen Analyse wird der Quelltext eingeteilt in Symbole wie Konstanten, Namen, Schlüsselwörter, Begrenzungszeichen und Operatoren. Kommentare werden eliminiert. Gleichzeitig werden Namenstabellen mit Variablen-, Konstanten-, Prozedurnamen oder ähnlichem aufgebaut.

Die syntaktische Analyse arbeitet nicht mehr mit einzelnen Zeichen, sondern mit den vom Scanner gelieferten Symbolen. Dabei wird die eigentliche Syntaxprüfung vorgenommen, also die richtige Stellung der Schlüsselwörter, Strichpunkte etc. Der Parser liefert auch die Basis für die nachfolgende Codeerzeugung.

Die Codeerzeugung liefert im Prinzip ein lauffähiges Maschinenprogramm, es heißt jetzt Objektcode und wird in einer Objektdatei abgelegt. Dieses Programm kann noch - möglicherweise vom Compiler selbst - optimiert und danach vom Linker zum endgültigen Programm verarbeitet werden. Doch wozu nun noch der Linker ?

Aufgaben des Linkers

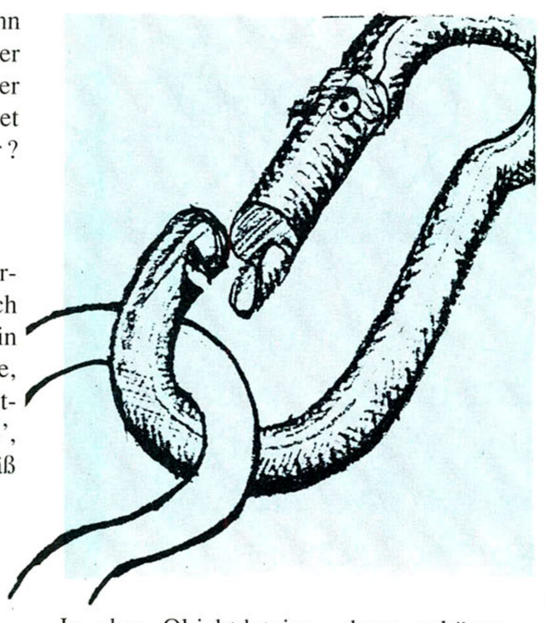
Wie gesagt, ist das vom Compiler erzeugte Programm für sich alleine noch nicht lauffähig. Es befinden sich darin beispielsweise Unterprogrammaufrufe, die noch keine richtige Sprungadresse enthalten. Schreibt man in C etwa 'printf(0)', so übersetzt der Compiler dies sinngemäß in folgendes Maschinenprogramm:

```
MOVE   #0,-(SP)
JSR    ...
ADDQ   #2,SP
```

Das heißt im Klartext: Bringe die '0' als Parameter auf den Stack, rufe dann das Maschinenprogramm 'printf' auf und bringe anschließend den Stack wieder in Ordnung. Nun weiß der Compiler aber noch gar nicht, wo das 'printf'-Programm sich im Speicher befinden wird, deshalb kann er den JSR-Befehl nicht mit einer vernünftigen Adresse versorgen. Stattdessen setzt er an dieser Stelle eine Mitteilung an den Linker ein, so daß dieser die fehlende Adresse ergänzen kann.

Erst der Linker weiß ja, wo sich das 'printf'-Programm letztendlich befindet, denn er verbindet das compilierte Programm mit den entsprechenden Libraries (Bibliotheken), wo unter anderem auch das 'printf'-Unterprogramm versteckt ist. Dasselbe passiert, wenn man in einem Programm externe Variablen verwendet, die in einem anderen Quelltext definiert sind. Auch hier sind die endgültigen Adressen nur dem Linker bekannt. Das wäre dann auch schon die wichtigste Aufgabe des Linkers, aber er tut noch etwas mehr.

Beim TOS werden Programme in drei Bereiche aufgeteilt, TEXT, DATA und BSS. Der TEXT-Bereich (oder TEXT-Section) sollte immer Maschinenbefehle enthalten; der DATA-Bereich (oder DATA-Section) sollte Daten enthalten, darunter versteht man vordefinierte Felder, Stringkonstanten und so weiter; der BSS-Bereich (-Section) schließlich enthält undefinierte Daten (leere Variablen oder Felder). Da die Daten im BSS-Bereich undefiniert sind, ist es nicht nötig, sie auf Diskette abzuspeichern. Das Betriebssystem reserviert beim Start des Programms den nötigen Speicherplatz und besetzt ihn einfach mit Nullen vor.



In den Objektdateien, dazu gehören natürlich auch die Bibliotheken, sind die Einträge in die verschiedenen Sections bunt durcheinandergewürfelt und müssen vom Linker sortiert werden. Am Ende entsteht also eine große TEXT-Section, eine große DATA-Section und eine große BSS-Section. Nun aber ins Detail, und damit zum GST-Format:

Das GST-Objectfile-Format

Das GST-Format entspricht gewissen Syntaxregeln, die vom Linker analysiert und überprüft werden. Der Linker arbeitet also genauso wie ein Compiler, nur die Syntaxregeln sind bedeutend einfacher. Diese Syntaxregeln sehen in Backus-Naur Form wie folgt aus:

```
<objectfile>      ::=  <module> { <module> }
<module>          ::=  <source> { <chunk> } <end>
<source>          ::=  $FB $01 <string>
<end>             ::=  $FB $13
<string>          ::=  <byte> { <byte> }
<chunk>           ::=  { <header_command> }[ <section_command> <body> ]
<header_command>  ::=  <comment> | <xdef> | <define>
<comment>         ::=  $FB $02 <string>
<xdef>            ::=  $FB $06 <string> <long> <id>
<define>          ::=  $FB $10 <id> <string>
<id>              ::=  <word>
<word>            ::=  <byte> <byte>
<long>            ::=  <word> <word>
<section_command> ::=  <section> | <org> | <common>
<section>         ::=  $FB $04 <id>
<org>             ::=  $FB $03 <long>
<common>          ::=  $FB $12 <id>
<body>            ::=  { <data_byte> | <body_command> }
<body_command>    ::=  <offset>  |  <xdef> | <xref> | <define>  |  <comment>
<offset>          ::=  $FB $05 <long>
<xref>            ::=  $FB $07 <long> <byte> { <op> <id> } $FB
<op>              ::=  + | -
<data_byte>       ::=  $00 | ... | $FA | $FB $FB | $FC |... | $FF
```

Die geschweiften Klammern deuten eine beliebige Wiederholung eines Elements an, auch nullmal ist erlaubt. Die eckige Klammer bedeutet null- oder einfaches Auftreten eines Elements. Der senkrechte Strich heißt "oder".

Man sieht, daß ein Objektfile aus beliebig vielen Modulen bestehen kann. Wenn man ein normales Programm compiliert oder assembliert, entsteht ein Objektfile, das genau ein Modul enthält. Eine Library besteht im allgemeinen aus mehreren Modulen. Man kann sich Libraries selbst erzeugen, indem man mehrere Objektfiles aneinanderhängt.

Ein Modul beginnt mit einem <source>-Kommando, gefolgt von beliebig vielen <chunk>s und endet mit einem <end>-Kommando. Zum <string> ist noch zu sagen, daß das erste Byte die Länge des Strings angibt und dann entsprechend viele Zeichen folgen. Innerhalb eines <chunk>s stehen neben Symboldefinitionen und Referenzen die eigentlichen Datenbytes für eine Section. Im GST-Format sind beliebig viele Sectionnamen erlaubt, nicht nur TEXT, DATA und BSS. Ein <comment> ist ein Kommentar, der keine weitere Bedeutung hat. Mit <xdef> wird ein Symbolname(<string>) definiert, wobei <long> den Wert des Symbols bezüglich des Sectionstarts der Section <id> angibt. Es wird hier nicht vom Namen der Section Gebrauch gemacht, sondern von einer <id>, die mit dem <define>-Kommando dem Namen zugeteilt wird. Diese Vorgehensweise soll das Auftauchen immer wieder gleicher Namen im Objektfile vermeiden und so Speicher sparen. Negative <id>s gehören zu Sectionnamen, positive <id>s zu normalen Symbolnamen und die <id> Null bedeutet im <xdef>-Kommando, daß ein absolutes Symbol definiert



Zwei oder lieber vier Megabyte RAM gefällig ?

! Voilà !

Passend für alle Atari Computer mit den Typenbezeichnungen 260 ST, 520 ST, 520 ST+, 520 STM, 1040 STF bieten wir Ihnen Arbeitsspeicher satt mit überzeugenden Qualitätsvorteilen :

- Die Erweiterung ist in **drei Ausbaustufen** erhältlich
- Der Einsatz von Stecksockeln ermöglicht jederzeit den **Ausbau auf bis zu 4 MB**
- Sie müssen **keine** Leiterbahnen bzw. Widerstände durchtrennen
- Die ausgeklügelte Einstecktechnik macht bei der Montage den **Lötkolben überflüssig**

4 MB Basisplatine mit 0 MB bestückt	DM 238.50
4 MB Basisplatine mit 2 MB bestückt	DM 930.-
4 MB Basisplatine mit 4 MB bestückt	DM 1699.90
passender Steckadapter für 1040 STF	DM 57.-
passender Steckadapter für alle übrigen Megabit-RAM's (16 Stück)	DM 64.-
	DM 798.-

Aufgrund des schnellen Wechsels am Speichermarkt können sich die Preise geändert haben; fragen Sie daher bitte die aktuellen Tagespreise telefonisch an.

!NEU MEGA - CLOCK NEU!

Echtzeituhr für alle "kleinen" ST

- funktionsidentisch zur Uhr im MEGA ST
- mit dem neuen Blitter - TOS ist der Betrieb ohne Software möglich
- Software für das alte TOS wird mitgeliefert
- langzeitbatteriegepuffert
- kann ebenfalls ohne Löten eingebaut werden
- moderne SMD-Technik sorgt für geringsten Platzbedarf

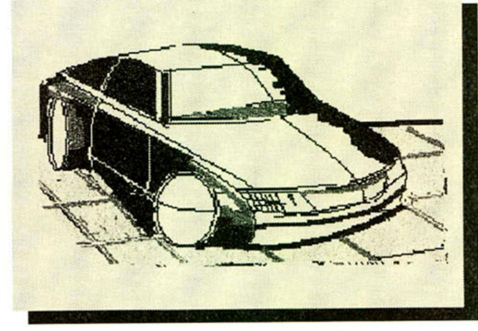
MEGA - Leistung , MINI - Preis DM 96.90

Die Lieferung erfolgt per Nachnahme zuzüglich DM 6,50 Versandkosten. Anfragen und Bestellungen richten Sie bitte an :



Kaiserstraße 5 - 7
5657 Haan 1
Tel. 02129 / 50819

CAMPUS ART





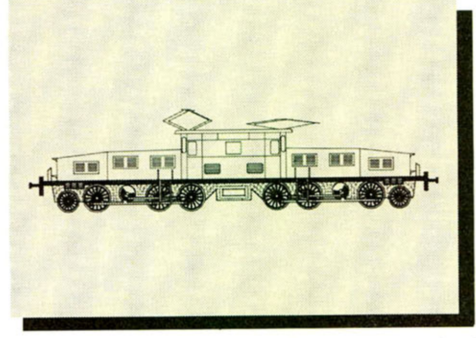


Das leistungsstarke Farbgraphiksystem für kreative Menschen!

- Arbeitstechniken wie im Graphicstudio (colorieren, schattieren, fotokopieren, ausschneiden, etc...)
- Eigenschaften der Stifte werden berücksichtigt
- Dreidimensionale Objektverarbeitung
- Benutzerfreundlichkeit durch Online Help

System: Atari ST, Farb-Bildschirm, Maus

CAMPUS DRAFT



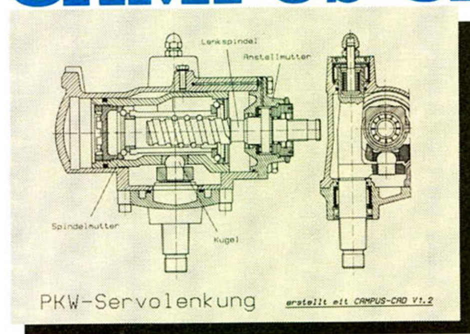


Vektororientiertes Zeichenprogramm mit allen Vorzügen professioneller CAD Systeme!

- Leichte Bedienung durch Maus, Menüs und Piktogramme
- Alle notwendigen Zeichenfunktionen (Lot, Parallele etc...)
- Vollständige Bemaßung, ausgereifte Symboltechnik
- Inclusive Drucker- und Plottertreiber

System: Atari ST, Monochrommonitor, Maus

CAMPUS CAD V1.2





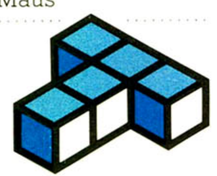
Neue Zusatzprogramme für das bewährte CAD-System!

- **CAMPUS ASC:** Konvertiert CAD ↔ ASCII **248,— DM**
- **CAMPUS PLT:** Konvertiert HPGL → CAD **248,— DM**
- **CAMPUS SYM:** Symbolbibliotheken für Elektronik, Elektrotechnik, Hydraulik/Pneumatik u.a. **248,— DM**
- Universeller Plottertreiber

System: Atari 1040ST, Monochrommonitor, Maus

Technobox

Kornharpener Str.122a · 4630 Bochum 1
Telefon: 02 34 / 50 30 60 · FAX: 50 30 61
CH: Senn Computer AG Zürich, Tel. 01/242 80 88
A: Steco Data, Dornbirn, Tel. 0 55 72/6 58 12

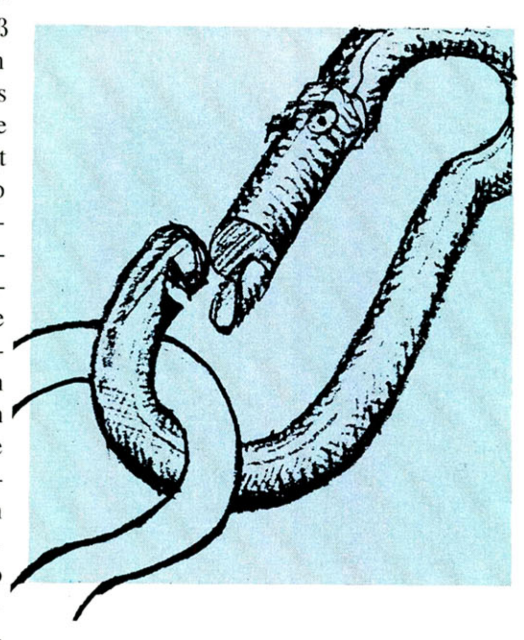


wird. Das <section>-Kommando legt fest, welcher Section die nachfolgenden Datenbytes zugeordnet werden sollen, es muß deshalb immer vor dem Auftauchen irgendwelcher Datenbytes stehen. Das <org>-Kommando bestimmt eine absolute Adresse, ab der die Datenbytes abgelegt werden. Das <common>-Kommando entspricht ziemlich genau dem <section>-Kommando, allerdings werden die folgenden Datenbytes nicht an die bereits vorhandene Section angehängt, sondern diese wird durch die neue Definition überschrieben und, falls der neue Bereich länger als der alte ist, verlängert. Mit dem <offset>-Kommando kann man eine bestimmte Position innerhalb der momentanen Section anwählen, von der ab dann die noch folgenden Datenbytes eingetragen werden. Das <xref>-Kommando trägt nun einen Symbolwert, beziehungsweise einen Wert der durch Additionen und Subtraktionen von Symbolwerten entsteht, an der augenblicklichen Position ein. Der Wert ergibt sich aus dem konstanten <long>-Wert und die durch die folgenden <id>s gegebenen Symbolwerte, die bei <op>='+' addiert und bei <op>='-' subtrahiert werden. Die spezielle <id> Null hat hier die Bedeutung der augenblicklichen Position im Gesamtprogramm. Das Ergebnis kann nach der Berechnung als Byte, Wort oder Langwort verwendet werden, außerdem kann es vorzeichenbehaftet und sogar noch relativ zur augenblicklichen Position sein. Über diese Möglichkeiten gibt das im <xdef>-Kommando enthaltene <byte> Auskunft, die Bits dieses Bytes haben, falls sie gesetzt sind, folgende Bedeutung:

```
Bit 0 :  Das Ergebnis ist ein Byte.
Bit 1 :  Das Ergebnis ist ein Wort.
Bit 2 :  Das Ergebnis ist ein Lang-
         wort.
Bit 3 :  Das Ergebnis ist vorzei-
         chenbehaftet.
Bit 4 :  Das Ergebnis ist vorzeich-
         enlos.
Bit 5 :  Das Ergebnis ist PC-relativ
         zu interpretieren.
Bit 6 :  Das Ergebnis benötigt zu-
         sätzliche Relozierung nach
         dem Laden, das heißt, es
         muß in eine Tabelle ein
         getragen  werden, die sich
         am Ende des Programm-Files
         befindet, und die das Be-
         triebssystem zur end-
         gültigen Anpassung des Pro-
         gramms an die endgültige
         Startadresse benötigt.
```

Dazu jetzt ein kleines Beispiel in C :

```
void main()
{
    printf(0);
}
```



Der LATTICE C-Compiler erzeugt daraus folgendes Programm:

```
42A7            CLR.L   -(SP)
4EB9 0000 0000  JSR     PRINTF
588F            ADDQ.L  #4,SP
4E75            RTS
```

Das zugehörige Objektfile sieht so aus:

```
FB 01 08 42 45 49 53 50 49 45 4C
    SOURCE BEISPIEL
FB 10 FF FF 04 74 65 78 74
    DEFINE -1 TEXT
FB 10 FF FE 04 64 61 74 61
    DEFINE -2 DATA
FB 10 FF FD 05 75 64 61 74 61
    DEFINE -3 UDATA
FB 10 FF FC 05 64 65 62 75 67
    DEFINE -4 DEBUG
FB 10 00 05 04 5F 33 32 6B
    DEFINE  5 _32K
FB 04 FF FF
    SECTION -1
FB 06 04 6D 61 69 6E 00 00 00 00 FF
                                 FF
    XDEF MAIN 0 -1
42 A7 4E B9
    Code für CLR.L -(SP) und JSR
FB 10 00 06 06 70 72 69 6E 74 66
    DEFINE  6 PRINTF
FB 07 00 00 00 00 54 2B 00 06 FB
    XREF 0 54 + 6
58 8F 4E 75
    Code für ADDQ.L #4,SP und RTS
FB 04 FF FE
    SECTION -2
FB 04 FF FD
    SECTION -3
FB 13
    END
```

Dazu folgende Erläuterungen: XDEF MAIN 0 -1 heißt, daß das Symbol MAIN den Wert 0 relativ zur Section -1 hat. Aus den vorangegangenen DEFINE-Kommandos erkennt man, daß -1 die TEXT-Section ist. Das Kommando XREF 0 54 + 6 heißt: Trage den Wert 0 plus den Wert des Symbols 6 als vorzeichenloses Langwort, mit Eintrag in die 'Runtime-Relocation-Table', an der momentanen Programmposition ein. Symbol 6 ist, wie man am darüberliegenden DEFINE erkennt, das PRINTF-Symbol. Damit ist der JSR-Befehl vollständig mit der richtigen Adresse versorgt. Die Sections DATA, UDATA und DEBUG, die zwar per DEFINE erwähnt sind, erhalten keine Daten. Doch nun zum Linker selbst:

Die Arbeitsweise des Linkers LINK

Der hier vorgestellte Linker verarbeitet alle Objektfiles in einem Durchlauf (Pass) und erstellt das lauffähige Programm im Speicher. Der dazu nötige Speicherbereich ist auf 100 Kbyte festgelegt, kann aber per Option (fast) beliebig vergrößert oder verkleinert werden. Nach Fertigstellung des Programms im Speicher wird dieses auf Datei geschrieben. Der Aufbau des Programms aus den Objektfiles geschieht wie folgt:
Für jedes Sprachelement des Objektfiles steht im Programm genau eine Prozedur, die eben dieses Sprachelement analysiert, dies sind die Prozeduren :

```
'link_file'        für  <object_file>
'module'           für  <module>
'chunk'            für  <chunk>
'body'             für  <body>
'section_command'  für <section_command>
'header_command'   für <header_command>
'xref_dir'         für  <xref>
'xdef_dir'         für <xdef>
'offset_dir'       für <offset>
'data_dir'         für <data_byte>
'common_dir'       für <common>
'org_dir'          für <org>
'section_dir'      für <section>
'comment_dir'      für <comment>
```

Diese Prozeduren rufen sich den Syntaxregeln entsprechend gegenseitig auf. Diese Methode findet sich auch in vielen Compilern wieder; ein Compiler, der nach diesem Prinzip arbeitet, heißt 'Recursive-Descent-Compiler'. Da wir nun schon bei der Analogie zum Compiler sind, sei noch folgendes erwähnt: Scanner, Parser und Codegenerator arbeiten ineinander verschränkt, der Linker benötigt also nur einen einzigen Pass. Der Scanner ist realisiert in der Prozedur 'nxsy', die die Bytefolgen aus dem Objektfile in syntaktische Einheiten zerlegt. Diese Einheiten spiegeln sich in der Typdefinition von 'DIRECT' wieder. Der Parser besteht genau aus den obengenannten Prozeduren, die nicht mehr mit den Datenbytes, sondern mit den vom Scanner gelieferten Einheiten arbeiten. Diese Prozeduren bewerkstelligen auch zum größten Teil die Codeerzeugung, das heißt, sie bringen die aus den Objektfiles ankommenden Sections in die endgültige Reihen-

folge. Gleichzeitig bauen sie ein nicht unerhebliches Tabellenwerk auf, das das Auffinden von Symbolen, Sections, Modulen und <xref>s ermöglicht.

Erst nach dem Einlesen sämtlicher Objektfiles sind (hoffentlich) alle Symbole definiert, und die aufgetretenen <xref>s können berechnet werden. Diese Berechnung geschieht in der Prozedur 'all_xrefs', die ihrerseits die Prozedure 'calc_xref' zur Bearbeitung eines <xref>s benutzt. In 'all_xrefs' entsteht auch die "Runtime-Relocation-Table", die Tabelle also, die es dem Betriebssystem ermöglicht, das Programm an jede beliebige Position im Speicher zu verschieben. Die Prozedur 'debug_table' hängt, falls gewünscht, an das Programmfile noch die gesamte Symboltabelle an, die von einem symbolischen Debugger weiterverwendet werden kann. Das fertige Programm wird von der Prozedur 'write_prog' auf Datei geschrieben.

Bei Libraries gibt es nun die Besonderheit, daß die in ihr enthaltenen Modulen nur hinzugelinkt werden, wenn dort Symbole definiert sind, die auch woanders gebraucht werden. Die Prozedur 'test_module' liest deshalb zunächst ein ganzes Modul in einen Pufferspeicher und stellt fest, ob eines der in diesem Modul definierten Symbole benötigt wird. Nur wenn das der Fall ist, wird der Pufferinhalt dem Programm in gewohnter Weise hinzugefügt. Die Größe des Puffers ist mit 32 Kbyte vordefiniert, kann aber per Option geändert werden.

Achtung! Prozeduren innerhalb eines Moduls kann der Linker nicht einzeln hinzufügen oder weglassen, denn innerhalb eines Moduls können sich die Prozeduren gegenseitig aufrufen, ohne daß der Linker davon etwas merkt. Definiert man etwa in einem C-Quelltext zwei Prozeduren, von denen eine die andere aufruft, so kann der Aufruf meist bereits vom Compiler durch einen relativen Sprung compiliert werden, der Linker hat an dieser Stelle dann keine Arbeit mehr.

Zum Aufbau und Suchen von Tabellen existieren folgende Prozeduren:

```
'app_mod' : Einfügen eines Modulnamens in die Modulnamentabelle;
            dies ist eine lineare Liste.
'app_xsy' : Einfügen eines Symbols in einen binären Baum.
'app_sec' : Einfügen einer Sectiondefinition in eine lineare Liste.
'src_xsy' : Suchen eines Symbols im Binärbaum.
'src_sec' : Suchen einer Sectiondefinition in der linearen Liste.
```

Der Vollständigkeit halber seien noch die folgenden Prozeduren erwähnt:

```
'halt'         : Abbruch des Programms nach einem fatalen Fehler.
'strnicmp'     : Vergleich zweier Strings ohne Unterscheidung von
                 Klein- und Großbuchstaben mit Längenbegrenzung.
'statistic'    : Anzeige der Sections mit Längenangabe
'list_xsy'     : Ausdrucken aller oder nur undefinierter Symbole.
'get_drct'     : Ein Byte aus dem Objektfile lesen.
'get_byte'     : Ein Byte entweder aus dem Modulpuffer oder dem
                 Objektfile lesen
'move_up'      : Speicherbereich nach oben schieben, um so Platz
                 für die Erweiterung einer Section zu schaffen
'init_mem'     : Speicherbereiche per 'malloc' reservieren
'make_ext'     : Falls ein Dateiname keine Extension besitzt, so
                 wird hier eine angehängt. Dazu werden die
                 Funktionen 'stcgfe' und 'strmfe' aus der Library
                 des LATTICE-C gebraucht. 'stcgfe' isoliert die
                 Extension aus einem Dateinamen, 'strmfe' fügt
                 eine neue Extension an.
'command_line' : Diese Prozedur analysiert die Kommandozeile und
                 setzt entsprechende Flags.
```

Weshalb überhaupt noch einen Linker für das GST-Format ?

Besitzer des GST-Linkers werden sich sicherlich schon oft über die mangelnde Geschwindigkeit desselben, aber auch über die immer wiederkehrenden Bomben während des Linkprozesses geärgert haben. Der hier vorgestellte Linker arbeitet bei Diskettenbetrieb mit der 4.5-fachen Geschwindigkeit und hat bisher auch bei größeren Programmen keine Bomben geworfen.

Ein weiterer Nachteil des GST-Linkers besteht darin, daß dieser nur die TEXT-Section des ablauffähigen Programms benutzt. Das heißt, daß alle in den Objektfiles definierten Sections dort landen. Bei der DATA-Section (initialisierte Felder, Stringkonstanten usw.) wäre das nicht weiter schlimm, wer in seinem Programm aber leere Felder von mehreren Kilobytes Größe vereinbart hat, wird sicher wenig Verständnis dafür haben, daß diese unnötigerweise Speicherplatz auf der Diskette belegen. Normalerweise sind nämlich nur TEXT- und DATA-Bereiche auf Datei abgelegt, der BSS-Bereich wird erst nach dem Laden des Programms vom Betriebssystem initialisiert. Dieser Linker verteilt die Bereiche richtig auf TEXT-, DATA- oder BSS-Section, so daß die ablauffähigen Programme nicht selten gewaltig zusammenschrumpfen. Dies kann aus Kompatibilitätsgründen an- und abgeschaltet werden.

Welche Sections wo landen, ist in der Prozedur 'def_section' festgelegt. Diese ordnet beliebigen Sectionnamen die Zahl -1 zu, so daß diese Sections in die TEXT-Section des TOS gelangen. Sections mit Namen 'DATA' wird die Kennung -2 zugeteilt, was zur Folge hat, daß diese auch in die DATA-Section des TOS gebracht werden. Sections mit Namen 'UDATA' und 'BSS' bekommen die Kennung -3 und werden dadurch der BSS-Section des TOS zugeordnet. Diese wird, im Gegensatz zur Vorgehensweise des Original-GST-Linkers, nicht auf Diskette abgespeichert, falls man die -SEC Option eingeschaltet hat. Die Namen sind kompatibel zum LATTICE C, dieser nennt die BSS-Section nämlich 'UDATA', und zum Metacomco Assembler, der schon den richtigen Namen 'BSS' verwendet . Wer noch andere Sections im BSS-Bereich untergebracht haben möchte, kann dies in 'def_section' in analoger Weise eintragen.

Aufruf des Linkers

Der Linker sollte als "TTP"-Programm benannt werden, denn er erwartet die Eingabe einer Kommandozeile, die der des GST-Linkers sehr ähnlich ist, und folgendes Aussehen haben sollte:

```
[<module>] [<control> [<listing> [<program>]]] {<option>}

module : ist der Dateiname einer Objekt-Datei, die gelinkt werden
         soll. Dieser Name wird vom Linker immer mit ".BIN" ergänzt.
control: ist der Name einer Control-Datei, in der Linkeranweisungen
         stehen.
listing: ist der Name einer List-Datei.
program: ist der Dateiname des fertigen Programms.
```

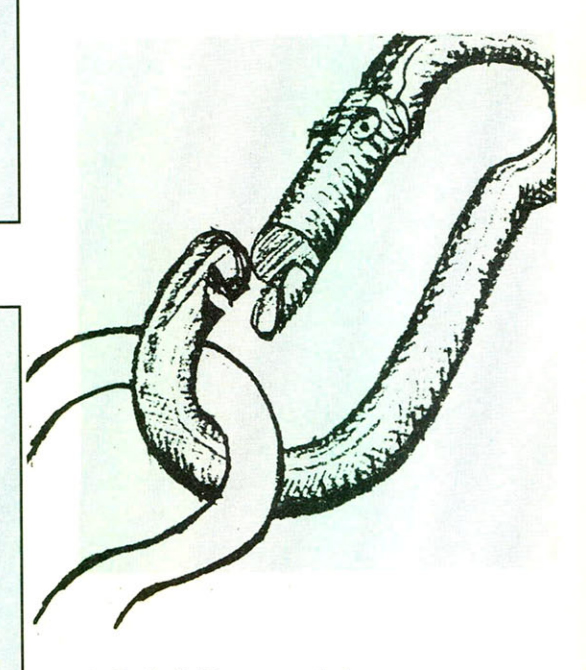
Als Options sind zulässig:

```
-LIST [<listing>]    :  Der Linker erzeugt ein Listing in der Datei
                        mit Namen <listing>.
-PROG [<program>]    :  Der Linker legt das Programm in der Datei
                        <program> ab.
-MEM <size>          :  Damit wird die Größe des Speichers in KByte
                        festgelegt in dem der Linker das fertige
                        Programm abspeichert. Also mit "-MEM 200"
                        können Programme bis zu 200 Kbyte Größe
                        gelinkt werden. Der Defaultwert ist
                        100 Kbyte.
-BUF <size>          :  Der Linker benutzt für die Bearbeitung von
                        Libraries einen Puffer, in dem er zunächst
                        Module ablegt, und dann testet, ob diese
                        dazugelinkt werden müssen oder nicht. Die
                        Größe des Puffers kann mit dieser Option in
                        Kbytes bestimmt werden. Der Defaultwert ist
                        32 Kbytes, und damit im allgemeinen
                        ausreichend.
-NOLIST              :  Der Linker produziert kein Listing.
-NODEBUG             :  Der Linker erzeugt keine Symboltabelle im
                        Programm.
-NOPROG              :  Der Linker erzeugt kein Programm.
-DEBUG               :  Das Programm erhält eine Symboltabelle.
-SYM                 :  Eine Symboltabelle wird ans Listing
                        angehängt.
-NOSYM               :  Keine Symboltabelle im Listing.
-SEC                 :  Sections mit Namen "TEXT" und andere werden
                        in der TEXT-Section abgelegt. Sections mit
                        Namen "DATA" werden in der DATA-Section
                        abgelegt. Sections mit dem Namen "UDATA"
                        oder
                        "BSS" werden in der BSS-Section abgelegt,
                        also nicht mit auf Diskette abgespeichert,
                        sondern beim Start des Programms mit Null
                        initialisiert.
-NOSEC               :  Alle Sections werden in der TEXT-Section
                        abgelegt, so wie das der GST-Linker tut.
-WITH [<control>]    :  Der Linker arbeitet mit dem Controlfile
                        <control>.
```

Im Controlfile können folgende Anweisungen stehen:

```
INPUT <filename>    : Die Datei <filename> wird in jedem Fall
                      hinzugelinkt.
LIBRARY <filename>  : Nur benötigte Module aus der Datei
                      <filename> werden hinzugelinkt.
```

In beiden Fällen wird an die Dateinamen die Extension ".BIN" angehängt, falls keine andere vorhanden ist.
Die Defaultwerte für die Options sind:
`-NOLIST -NODEBUG -NOSYM -NOSEC -MEM 100 -BUF 32`



Die Defaultfilenamen sind:

```
Input-file     : <module>.BIN
Listing-file   : <module>.MAP
Control-file   : <module>.LNK
Program-file   : <module>.PRG
```

Der <module>-Dateiname ist also immer notwendig.
Schließlich und endlich nun noch die Fehlermeldungen, die erscheinen können:

- **Error 0** : Out of memory
  Beim Aufruf der malloc-Funktion konnte der gewünschte Speicher nicht zur Verfügung gestellt werden. Die einzige Abhilfe wäre hier, den Computer ohne Accessories zu starten, um mehr Speicher zu haben.
- **Error 1** : Program memory too small
  Das Programm ist größer als 100 Kbyte, beziehungsweise als der mit der SEC Option vereinbarte Speicher. Abhilfe: Mit der -SEC Option mehr Speicher reservieren.
- **Error 2** : Error in binary File
  Das Objektfile entspricht nicht dem GST-Format. Leider hat der Metacomco-Assembler die Eigenart, unter gewissen Umständen fehlerhafte Objektfiles zu produzieren. Im allgemeinen hilft es, in bestimmten Datenbereichen (etwa bei Stringkonstanten) zusätzliche Bytes einzufügen. Genaueres läßt sich hier leider nicht sagen.
- **Error 3** : Too many operands in XREF
  In einem XREF-Kommando können maximal 10 Symbole in eine Berechnung eingehen, sind es einmal mehr, so

erscheint diese Fehlermeldung. In der Praxis dürfte das allerdings kaum vorkommen. Trotzdem Abhilfe: Konstante XMAX im Quelltext erhöhen und den Linker neu kompilieren.

- **Error 4** : Illegal section id
  Im einem Modul ist die Anzahl verschiedener Sections auf 20 begrenzt (sinnvoll sind sowieso nur drei). Sollten es dennoch mehr sein, so erscheint diese Meldung. Abhilfe: Ändern der Konstanten MAX_NDEF und neucompilieren des Linkers.
- **Error 5** : Illegal symbol id
  In einem Modul (nicht im gesamten Programm!) ist die Anzahl verschiedener Symbole auf 500 begrenzt. Sollten es dennoch mehr sein, so erscheint diese Meldung. Abhilfe: Ändern der Konstanten MAX_PDEF und neucompilieren des Linkers.
- **Error 6** : ORG encountered
  Das ORG-Kommando in einem Objektfile wird mangels Bedarf nicht vom Linker unterstützt.
- **Error 7** : Should not occur
  Taucht nie auf.
- **Error 8** : Word or longword at odd address
  Dies ist ein Fehler im Objektfile, für den der Compiler oder Assembler verantwortlich ist.
- **Error 9** : Buffer too small
  Der Puffer, der zum Zwischenspeichern für Module aus den Libraries gebraucht wird, ist zu klein. Abhilfe: mit der -BUF Option mehr Speicher reservieren.
- **Error 10** : Cannot write file correctly
  Das Programmfile kann nicht fehlerfrei geschrieben werden. Im allgemeinen ist dann kein Platz mehr auf der Diskette.

Außer diesen fatalen Fehlern werden natürlich undefinierte oder mehrfach definierte Symbole gemeldet, dies führt aber nicht zum Abbruch des Programms.

Sollte das Linken nicht geklappt haben, meldet der Linker dem aufrufenden Programm einen Fehlercode 1, ansonsten Null. Manche Shell-Programme (wie z.B. das "MENU+" von Metacomco) machen Gebrauch davon.

Ingo Eichenseher

ENDE


```
/* Linker Version vom 08.01.1988 04:15 */
/* Ingo Eichenseher, Am Leiterle 9, D-8901 Stadtbergen
 */

#include <stdio.h>
#include <osbind.h>
#include <string.h>
#include <ctype.h>

#define FMSIZE     64
#define MAX_LEN    32
#define MAX_PDEF  500
#define MAX_NDEF   20
#define BLEN     1024
#define XMAX       10

#define sgn(x) ((x)<0?-1:((x)==0?0:1))

typedef enum direct { data,source,comment,org,section,
                      offset,xdef,xref,
                      define,common,end,eofsy } DIRECT;

typedef char ALFA[MAX_LEN];

typedef struct oper
    {
        short id;
        char  op;
    } OPER;

typedef struct symbol
    {
        short          length;
        DIRECT         directive;
        char           string[81];
        long           longword;
        short          id;
        char           trunc_rule;
        unsigned char  data_byte;
        short          n_xref;
        OPER           xref_oper[XMAX];
    } SYMBOL;

typedef struct mod_item
    {
        struct mod_item   *mod_next;
        char               mod_name[2];
    } MOD_ITEM;

typedef struct section
    {
        struct section *sec_next;
        short           sec_id;
        char           *sec_start;
        long            sec_length;
        long            sec_oldlen;
        long            sec_fxref;
        long            sec_xptr;
        MOD_ITEM       *sec_module;
        char            sec_name[MAX_LEN];
    } SECTION;

typedef struct xsym
    {
        SECTION        *xsy_sect;
        MOD_ITEM       *xsy_mod;
        long            xsy_value;
        short           xsy_defd;
        struct    xsym *xsy_left;
        struct    xsym *xsy_right;
        char            xsy_name[MAX_LEN];
    } XSYMBOL;

typedef struct xoper
    {
        char      xop_oper;
        char      xop_optyp;
        union
            {
                SECTION  *xop_sec;
                XSYMBOL  *xop_sym;
            } xop_ptr;
    } XOPER;

typedef struct xref
    {
        struct xref    *xref_next;
        long            xref_pos;
        long            xref_abs;
```

- ★ kurze Einarbeitungszeit
- ★ dialogorientiert
- ★ menügesteuert
- ★ GEM Oberfläche
- ★ übersichtliche Eingabebildschirme
- ★ kein Hardware-Kopierschutz, kein Port belegt
- ★ **Auswertungen jederzeit in Sekundenschnelle auf Bildschirm, Drucker oder Massenspeicher**
- ★ frei wählbares Wirtschaftsjahr
- ★ variable Mwst.-Sätze
- ★ **integriertes Abschreibungsprogramm**
- ★ Journal, Saldenliste
- ★ Kassenbuch, Finanzkonten
- ★ Kosten- und Erlöselisten
- ★ Wareneingangsliste
- ★ **USTvoranmeldung direkt auf das amtliche Formular**
- ★ Einnahme-Überschuß-Rechnung (fibuMAN e/m)
- ★ **Bilanz und Gewinn- und Verlustrechnung nach dem Bilanzrichtliniengesetz (fibuMAN f/m)**
- ★ BWA – Betriebswirtschaftliche Auswertung (fibuMAN m, optional fibuMAN f)
- ★ Sammeldruck aller zum Monatsabschluß relevanter Auswertungen
- ★ Kontoauszüge über das gesamte Jahr nach Konten chronologisch geordnet
- ★ Hinweis auf unsinnige Buchungen
- ★ **auf Mausklick automatisches Anlegen einer Stornobuchung**
- ★ Automatikjournal zur einmaligen Eingabe monatlich wiederkehrender Buchungen
- ★ diverse Schnittstellen (fibuSTAT, Faktura in Vorbereitung: faktuMAN)
- ★ schnelles Suchen nach Buchungen mit optionaler Übernahme in das aktuelle Journal, z. B. Ausbuchungen von offenen Posten
- ★ Kontenanlage mit optionalem Automatiktext und -UST auch während der Buchungen
- ★ Hochrechnung der Mwst. von Nettobeträgen
- ★ Kennwort auf 2 Ebenen
- ★ Sicherheitskopien nach frei wählbarem Zeitraum
- ★ **komfortable Druckeranpassung**
- ★ Ordner vom Programm aus anzulegen
- ★ Dateien vom Programm aus zu löschen

Ich bin mit keinem Buchhaltungs-Programm zufrieden!!!

Ich brauche Zuverlässigkeit, Schnelligkeit, Übersichtlichkeit, aussagekräftige, gesetzlich anerkannte Auswertungen

und habe keine Zeit, mich erst lange einarbeiten zu müssen...



fibuMAN

BY H.-G. SIEGEL

DER FINANZBUCHHALTUNGS-MANAGER

NEU VERSION 2.0

Siehe Testbericht ST-Computer 12/87

fibuMAN m	fibuMAN f	fibuMAN e	BWA ZU fibuMAN f DM 98,-
MANDANTENFÄHIGE FIBU MIT BWA	FINANZ-BUCHHALTUNG	EINNAHME-ÜBERSCHUSS-RECHNUNG	DEMODISKETTE MIT HANDBUCH
DM 968,-	DM 768,-	398,-	DM 65,-

Preise für MS Dos Versionen auf Anfrage
PRODATA bietet Ihnen vollen professionellen Anwenderservice wie kostenlose telefonische Hotline, täglich von 10 – 23 Uhr, Schulung und individuelle Einarbeitung, ggf. Anpassung der Programme an Ihre Bedürfnisse, prompten Updateservice und, und, und...
Ein Aufstieg in der fibuMAN-Reihe ist jederzeit unter Anrechnung des Preises der ursprünglichen Version möglich. Sie können alle hier aufgeführten Produkte direkt beim Entwickler PRODATA oder im autorisierten Fachhandel erwerben.

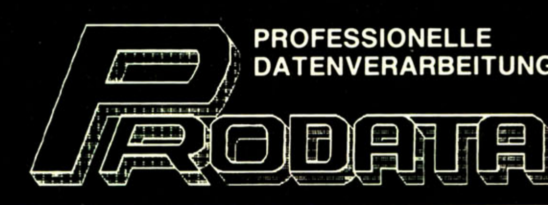
fibuKURS . . . . DM 348,-

- ★ Diskette mit ausführlichem Lehrbuch (weit über 100 Seiten)
- ★ leichter Einstieg in eine eigene EDV-Buchführung
- ★ Finanzbuchhaltung und Einnahme-Überschuß-Rechnung
- ★ Nachschlagewerk für das spätere praktische Buchen
- ★ Bibliothek von weit über 100 anschaulichen Fallbeispielen
- ★ nach Abschluß des Kurses ist selbständiges Arbeiten möglich

fibuSTAT . . . . DM 398,-

- ★ unabhängiges GEM-Statistikmodul zu allen fibuMAN Programmen
- ★ Berechnung von betriebswirtschaftlich interessanten Kennziffern
- ★ diverse grafisch vergleichende Statistiken
- ★ übersichtliche Grafiken für einen schnellen Einblick in Ihre jeweilige Finanzlage
- ★ Balken- und Kuchendiagramme

BROICHERSTRASSE 39
5060 BERGISCH GLADBACH 1
TEL. 0 22 04 / 5 14 56 – 01 61 / 221 57 91



ENTWICKLUNG UND VERTRIEB INTELLIGENTER ANWENDERSYSTEME

```
        short           xref_trunc;
        short           xref_ops;
        struct xoper    xref_oper[XMAX];
    } XREF;

extern void         app_xsy(XSYMBOL**,XSYMBOL*);
extern XSYMBOL*     src_xsy(XSYMBOL*,char*);
extern void         app_sec(SECTION**,SECTION*);
extern SECTION*     src_sec(SECTION*,char*);
extern int          read_b();
extern void         printsy();
extern void         nxsy();
extern void         move_up(SECTION*);
extern SECTION*     def_section(SECTION*,char*,short);
extern long         calc_xsy(XSYMBOL*);

SYMBOL    sy;
char      message[]="68000 GST-Format-Linker Version 2.4
                /8.1.1988\n\n";
char      module_name[80];
ALFA      *pdef_name;
ALFA      *ndef_name;
char      input_name[FMSIZE],file_name[FMSIZE],
          control_name[FMSIZE];
char      listing_name[FMSIZE],program_name[FMSIZE];
short     control_flag,listing_flag,program_flag,
          debug_flag,symbol_flag;
short     spar_flag;
char      *membot,*memtop,*memstart,*memend,*altstart,
          *code_ptr,*neustart;
char      *altxref,*debug_start,*debug_end;
long      mem_size,buf_size;
unsigned char *module_buffer,*module_end,*module_ptr,
              *module_top,*module_max;
SECTION   *curr_sec,*sec_liste,*moved_sec;
SECTION   **sec_lptr=&sec_liste;
MOD_ITEM  *mod_liste,*curr_mod;
XSYMBOL   *xsy_liste;
XREF      *xref_liste;
FILE      *list_file;
int       undefd_sym,double_sym,range_err;
short     header[14]={0x601a,0,0,0,0,0,0,0,0,0,0,0,0,0};

char      *errmsg[]= {
    "Out of memory",                        /* 0 */
    "Program memory too small",             /* 1 */
    "Error in binary file",                 /* 2 */
    "Too many operands in XREF",            /* 3 */
    "Illegal section id",                   /* 4 */
    "Illegal symbol id",                    /* 5 */
    "ORG encountered",                      /* 6 */
    "Should not occur",                     /* 7 */
    "Word or longword at odd address",      /* 8 */
    "Buffer too small",                     /* 9 */
    "Cannot write file correctly",          /* 10 */
};

void halt(n)
int n;
{
    printf("Error %2d: %s.\n",n,errmsg[n]);
    printf("Press any key to continue\n");
    gemdos(1);
    exit(1);
}

int       strnicmp(x,y,n)
register char       *x,*y;
register unsigned int       n;
{
    if (n<1) return(0);
    while(toupper(*x)==toupper(*y) && *x && *y && --n)
      { x++; y++; }
    return((int)(toupper(*x)-toupper(*y)));
}

void      statistic()
{
    SECTION  *s;
    fprintf(list_file,
    "\n-----------------\n");
    fprintf(list_file,"SECTION        START    LENGTH\n");
    fprintf(list_file,"-----------------\n");
    for (s=sec_liste; s!=NULL; s=s->sec_next)
      fprintf(list_file,
        "%-9s %8X %8X\n",s->sec_name,
        s->sec_start-membot,s->sec_length);
    fprintf(list_file,"-----------------\n");
}

MOD_ITEM *app_mod(mod_liste,name)
MOD_ITEM **mod_liste;
char      *name;
{
    MOD_ITEM *new;

    new=(MOD_ITEM*)malloc(sizeof(MOD_ITEM)+strlen(name));
    if (new==NULL) halt(0);
    strcpy(new->mod_name,name);
    new->mod_next=*mod_liste;
    *mod_liste=new;
    return(new);
}

void      app_xsy(xsy_liste,xsy_neu)
XSYMBOL   **xsy_liste,*xsy_neu;
{
    int   c;
    if (*xsy_liste==NULL)
    {
        *xsy_liste=xsy_neu;
        xsy_neu->xsy_left=xsy_neu->xsy_right=NULL;
    }
    else
        if ((c=stricmp(xsy_neu->xsy_name,
        (*xsy_liste)->xsy_name))<0)
              app_xsy(&(*xsy_liste)->xsy_left,xsy_neu);
        else  app_xsy(&(*xsy_liste)->xsy_right,xsy_neu);
}

XSYMBOL   *src_xsy(xsy_liste,name)
XSYMBOL   *xsy_liste;
char      *name;
{
    int   c;
    if (xsy_liste==NULL) return(NULL);
    if ((c=stricmp(name,xsy_liste->xsy_name))==0) return
    (xsy_liste);
    if (c<0) return(src_xsy(xsy_liste->xsy_left,name));
    else     return(src_xsy(xsy_liste->xsy_right,name));
    return(NULL);
}

long      calc_xsy(s)
XSYMBOL* s;
{
    long  value;

    value=s->xsy_value;
    if (s->xsy_sect!=NULL) value+=s->xsy_sect
    ->sec_start-membot;
    return(value);
}

void      debug_table(x)
XSYMBOL   *x;
{
    register char  *p;
    register short  i;
    if (x!=NULL)
    {
        if (code_ptr+14>=memtop) halt(1);
        debug_table(x->xsy_left);
        p=x->xsy_name;
        for (i=8; i--;){ *code_ptr++=*p; if (*p) p++; }
        if (x->xsy_sect!=NULL) *((short*)code_ptr)=0xA200;
        else *((short*)code_ptr)=0xA000;
        code_ptr+=2;
        *((long*)code_ptr)=calc_xsy(x);
        code_ptr+=4;
        debug_table(x->xsy_right);
    }
}

void      list_xsy(xsy_liste,u_flag)
XSYMBOL   *xsy_liste;
int       u_flag;
{
    if (xsy_liste!=NULL)
    {
        list_xsy(xsy_liste->xsy_left,u_flag);
        if (u_flag)
        {
            if (!(xsy_liste->xsy_defd&1))
            {
             fprintf(list_file,"Undefined Symbol: '%s'\n",
             xsy_liste->xsy_name);
             undefd_sym++;
```

```
            }
        }
        else
        {
            if (xsy_liste->xsy_defd&1)
            {
                fprintf(list_file,
                    "%-20s %08X%c",xsy_liste->xsy_name,
                    calc_xsy(xsy_liste),
                        xsy_liste->xsy_defd&2 ? '':'*');
                if (xsy_liste->xsy_sect!=NULL)
                    fprintf(list_file," %15s",
                    xsy_liste->xsy_sect->sec_name);
                else fprintf(list_file," %20s"," ");
                if (xsy_liste->xsy_mod!=NULL)
                    fprintf(list_file," %15s",
                    xsy_liste->xsy_mod->mod_name);
                fprintf(list_file,"\n");
            }
            else fprintf(list_file,"%-20s undefined\n",
                xsy_liste->xsy_name);
        }
        list_xsy(xsy_liste->xsy_right,u_flag);
    }
}

void    app_sec(sec_liste,sec_neu)
SECTION **sec_liste,*sec_neu;
{
    if (*sec_liste==NULL)
    {
        *sec_liste=sec_neu;
        sec_neu->sec_next=NULL;
    }
    else if (sec_neu->sec_id<=(*sec_liste)->sec_id)
            app_sec(&(*sec_liste)->sec_next,sec_neu);
        else
        {
            sec_neu->sec_next=*sec_liste;
            *sec_liste=sec_neu;
        }
}

SECTION* src_sec(sec_liste,name)
SECTION  *sec_liste;
char     *name;
{
    while(sec_liste!=NULL)
    {
        if (stricmp(sec_liste->sec_name,name)==0)
            return(sec_liste);
        sec_liste=sec_liste->sec_next;
    }
    return(NULL);
}

unsigned char  inp_buf[BLEN],*buf_ptr,*buf_end;
int            inp_hnd;
int    read_b()
{
    if (buf_end-inp_buf<BLEN) return(-1);
    buf_end=inp_buf+Fread(inp_hnd,BLEN,inp_buf);
    buf_ptr=inp_buf;
    return(buf_ptr<buf_end ? (int)*buf_ptr++ : -1);
}

int    get_drct()
{
    return( buf_ptr<buf_end ?
    (int)*buf_ptr++ : read_b());
}

int    get_byte()
{
    return(module_ptr<module_end ? (int)*module_ptr++ :
    get_drct() );
}

void  nxsy()
{
    int    c;
    if ((c=get_byte())==0xFB)
    {
        register char  *p;
        register int    i;
        switch(get_byte())
        {
            case -1    : sy.directive=eofsy;
                         sy.length=0;
                         break;
            case 0x01  : sy.directive=source;
                         sy.length=0;
                         p=sy.string;
                         for(i=get_byte();i-;)
                         *p++=get_byte();
                         *p='\0';
                         break;
            case 0x02  : sy.directive=comment;
                         sy.length=0;
                         p=sy.string;
                         for(i=get_byte();i-;)
                         *p++=get_byte();
                         *p='\0';
                         break;
            case 0x03  : sy.directive=org;
                         sy.length=0;
                         sy.longword=(get_byte()<<24)
                         +(get_byte()<<16)+
                            (get_byte()<<8)+(get_byte());
                         break;
            case 0x04  : sy.directive=section;
                         sy.length=0;
                         sy.id=(get_byte()<<8)+(get_byte())
                         break;
            case 0x05  : sy.directive=offset;
                         sy.length=0;
                         sy.longword=(get_byte()<<24)
                         +(get_byte()<<16)+
                            (get_byte()<<8)+(get_byte());
                         break;
            case 0x06  : sy.directive=xdef;
                         sy.length=0;
                         p=sy.string;
                         for(i=get_byte();i-;)
                         *p++=get_byte();
                         *p='\0';
                         sy.longword=(get_byte()<<24)+
                         (get_byte()<<16)+
                            (get_byte()<<8)+(get_byte());
                         sy.id=(get_byte()<<8)+(get_byte())
                         break;
            case 0x07  : sy.directive=xref;
                         sy.longword=(get_byte()<<24)+
                         (get_byte()<<16)+
                            (get_byte()<<8)+(get_byte());
                         sy.trunc_rule=get_byte();
                         sy.length=sy.trunc_rule&7;
                         for (i=0; (c=get_byte())!=
                         0xFB && i<XMAX; i++)
                         {
                            if (c!='+' && c!='-')
                            {  printf("Illegal XREF
                            Operator:%c\n",c); halt(2); }
                            sy.xref_oper[i].op=c;
                            sy.xref_oper[i].id=
                            (get_byte()<<8)+(get_byte());
                         }
                         if (c!=0xFB) halt(3);
                         sy.n_xref=i;
                         break;
            case 0x10  : sy.directive=define;
                         sy.length=0;
                         sy.id=(get_byte()<<8)+(get_byte())
                         p=sy.string;
                         for(i=get_byte();i-;)
                         *p++=get_byte();
                         *p='\0';
                         break;
            case 0x12  : sy.directive=common;
                         sy.length=0;
                         sy.id=(get_byte()<<8)+(get_byte())
                         break;
            case 0x13  : sy.directive=end;
                         sy.length=0;
                         break;
            case 0xFB  : sy.directive=data;
                         sy.length=1;
                         sy.data_byte=0xFB;
                         break;
            default    : printf("Illegal Directive\n");
                         halt(2);
                         break;
        }
    }
    else
    {
        if (c==-1)
        {
```

```
            sy.directive=eofsy;
            sy.length=0;
        }
        else
        {
            sy.directive=data;
            sy.data_byte=c;
            sy.length=1;
        }
    }
}

void    move_up(s)
SECTION *s;
{
    if (memend!=memtop) halt(-1);
    moved_sec=s;
    if (s!=NULL)
        if (s->sec_start!=NULL)
        {
            altstart=memstart;
            memstart=s->sec_start;
            memend=memtop-(altstart-memstart);
            if (altstart>memstart) movmem(memstart,memend,
            altstart-memstart);
        }
        else moved_sec=NULL;
}

void  comment_dir()
{
    fprintf(list_file,"COMMENT: %s\n",sy.string);
    nxsy();
}

void  xdef_dir(body_flag)
int   body_flag;
{
    XSYMBOL  *s;

    if ((s=src_xsy(xsy_liste,sy.string))==NULL)
    {
        if ( (s=(XSYMBOL*)malloc(sizeof(XSYMBOL)))==
        NULL) halt(0);
        strupr(sy.string);
        strncpy(s->xsy_name,sy.string,MAX_LEN-1);
        s->xsy_defd=0;
        s->xsy_mod=NULL;
        app_xsy(&xsy_liste,s);
    }
    if (s->xsy_defd&1)
    {
        fprintf(list_file,"Double defined Symbol: %s\n",
        sy.string);
        double_sym++;
    }
    else
    {
        if (sy.id)
        {
            if (sy.id>0 || -sy.id>MAX_NDEF) halt(4);
            if ( (s->xsy_sect=src_sec(sec_liste,
            ndef_name[-sy.id]))==NULL)
            {
                if (body_flag) halt(2);
                s->xsy_sect=def_section(sec_liste,
                ndef_name[-sy.id],sy.id);
            }
        }
        else s->xsy_sect=NULL;
        s->xsy_value = sy.longword;
        if (s->xsy_sect!=NULL) s->xsy_value +=
        s->xsy_sect->sec_oldlen;
        s->xsy_defd  |=  1;
        s->xsy_mod = curr_mod;
    }
    nxsy();
}

void  define_dir()
{
    strupr(sy.string);
    if (sy.id>0)
    {
        if (sy.id>MAX_PDEF) halt(4);
        strncpy(pdef_name[sy.id],sy.string,MAX_LEN-1);
    }
    else
    {
        if (-sy.id>MAX_NDEF) halt(5);
        strncpy(ndef_name[-sy.id],sy.string,MAX_LEN-1);
    }
    nxsy();
}

SECTION  *def_section(sec_liste,name,id)
SECTION  *sec_liste;
char     *name;
short     id;
{
    SECTION  *sec;

    if (NULL==(sec=(SECTION*)malloc(sizeof(SECTION))))
    halt(0);
    strupr(name);
    strncpy(sec->sec_name,name,MAX_LEN-1);
    sec->sec_start  =  NULL;
    sec->sec_length =  0;
    sec->sec_oldlen =  0;
    sec->sec_id     =  -1;
    sec->sec_module = curr_mod;
    if (!stricmp(sec->sec_name,"DATA"))sec->sec_id=-2;
    if (!stricmp(sec->sec_name,"BSS"))sec->sec_id=-3;
    if (!stricmp(sec->sec_name,"UDATA"))sec->sec_id=-3;
    sec->sec_fxref  =  NULL;
    sec->sec_xptr   =  (long)&sec->sec_fxref;
    app_sec(sec_lptr,sec);
    return(sec);
}

void  sec_com_dir()
{
    if (sy.id>=0 || -sy.id>=MAX_NDEF) halt(4);
    if (NULL==(curr_sec=src_sec(sec_liste,
    ndef_name[-sy.id])))
    {
        curr_sec=def_section(sec_liste,
        ndef_name[-sy.id],sy.id);
        move_up(curr_sec->sec_next);
        curr_sec->sec_start=memstart;
    }
    else
    {
        move_up(curr_sec->sec_next);
        if (curr_sec->sec_start==NULL) curr_sec->
        sec_start=memstart;
    }
}

void  section_dir()
{
    sec_com_dir();
    code_ptr=neustart=memstart;
    nxsy();
}

void  org_dir()
{
    halt(6);
    nxsy();
}

void  common_dir()
{
    sec_com_dir();
    neustart=memstart;
    code_ptr=curr_sec->sec_start;
    nxsy();
}

void  data_dir()
{
    if (code_ptr>=memend) halt(1);
    *code_ptr++=sy.data_byte;
    nxsy();
}

void  offset_dir()
{
    if (code_ptr>neustart) neustart=code_ptr;
    code_ptr=memstart+sy.longword;
    if (code_ptr>neustart) neustart=code_ptr;
    nxsy();
}

void  xref_dir()
{
    XREF      *x;
```

```
    XSYMBOL  *xsy;
    SECTION  *sec;
    short     i,xid;

    if ((x=(XREF*)malloc(sizeof(XREF)+(sy.n_xref-XMAX)
    *sizeof(XOPER)))==NULL)
       halt(0);
    if (curr_sec==NULL) halt(2);
    x->xref_pos     =  code_ptr-curr_sec->sec_start;
    x->xref_abs     =  sy.longword;
    x->xref_ops     =  sy.n_xref;
    x->xref_trunc   =  sy.trunc_rule;
    x->xref_next    =  NULL;
    for (i=0; i<sy.n_xref; i++)
    {
       x->xref_oper[i].xop_oper=sy.xref_oper[i].op;
       xid=sy.xref_oper[i].id;
       switch(x->xref_oper[i].xop_optyp=sgn(xid))
       {
          case -1 : if (-xid>MAX_NDEF) halt(4);
                    if ( (sec=src_sec(sec_liste,
                    ndef_name[-xid]))==NULL)
                       sec=def_section(sec_liste,
                       ndef_name[-xid],xid);
                    x->xref_oper[i].xop_ptr.xop_sec=sec;
                    if (x->xref_oper[i].xop_oper=='+')
                          x->xref_abs += sec->sec_oldlen;
                    else x->xref_abs -= sec->sec_oldlen;
                    break;
          case  0 : break;
          case  1 : if ( (xsy=src_xsy(xsy_liste,
                    pdef_name[xid]))==NULL)
                    {
                      if ( (xsy=(XSYMBOL*)malloc(sizeof
                      (XSYMBOL)))==NULL )
                         halt(0);
                      strncpy(xsy->xsy_name,
                      pdef_name[xid],MAX_LEN-1);
                      xsy->xsy_defd=0;
                      xsy->xsy_mod=NULL;
                      app_xsy(&xsy_liste,xsy);
                    }
                    xsy->xsy_defd |= 2;
                    x->xref_oper[i].xop_ptr.xop_sym=xsy;
                    break;
       }
    }
    *((XREF**)curr_sec->sec_xptr)=x;
    curr_sec->sec_xptr=(long)&x->xref_next;
    code_ptr += sy.trunc_rule & 7;
    if (code_ptr>=memend) halt(1);
    nxsy();
}

void  header_command()
{
    int    in_header_com=1;

    while(in_header_com)
       switch (sy.directive)
       {
          case comment : comment_dir(); break;
          case xdef    : xdef_dir(0); break;
          case define  : define_dir(); break;
          default      : in_header_com=0; break;
       }
}

void  section_command()
{
    switch(sy.directive)
    {
       case section : section_dir(); break;
       case org     : org_dir(); break;
       case common  : common_dir(); break;
       default      : halt(2); break;
    }
}

void  body()
{
    while(sy.directive==data    || sy.directive==offset ||
       sy.directive==xdef   || sy.directive==xref ||
       sy.directive==define || sy.directive==comment )
    {
       switch(sy.directive)
       {
          case data    : data_dir(); break;
          case offset  : offset_dir(); break;
          case xdef    : xdef_dir(1); break;
          case xref    : xref_dir(); break;
          case define  : define_dir(); break;
          case comment : comment_dir(); break;
          default      : halt(2); break;
       }
    }
}

void  chunk()
{
    SECTION *s;

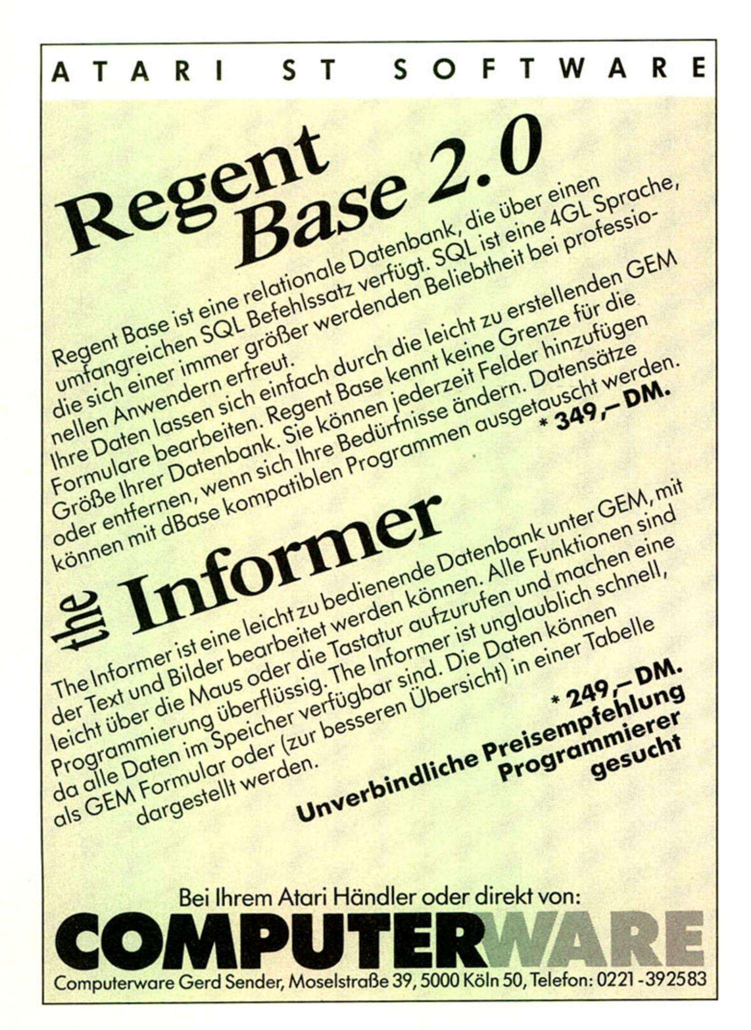
    while (sy.directive==xdef || sy.directive==comment ||
           sy.directive==define) header_command();
    if (sy.directive==section || sy.directive==org      ||
        sy.directive==common )
    {
       section_command();
       body();
       if ( ((long)code_ptr&1) && code_ptr>=neustart)
       {
          if (code_ptr>=memend) halt(1);
          *code_ptr++='\0';
       }
       if (code_ptr>neustart) neustart=code_ptr;
       curr_sec->sec_length+=neustart-memstart;
       if (altstart!=NULL)
       {
          if (altstart>memstart) movmem(memend,neustart,
          altstart-memstart);
          for(s=moved_sec; s!=NULL; s=s->sec_next)
             if (s->sec_start!=NULL) s->
             sec_start += neustart-memstart;
          memend=memtop;
          neustart+=altstart-memstart;
          altstart=NULL;
       }
       memstart=neustart;
    }
}

void  module()
{
    SECTION  *sec;
    short     i;

    if (sy.directive!=source) halt(2);
    curr_mod=app_mod(&mod_liste,sy.string);
    strcpy(module_name,sy.string);
    nxsy();
    while (sy.directive==xdef || sy.directive==comment ||
       sy.directive==define || sy.directive==section ||
       sy.directive==org     || sy.directive==
       common) chunk();
    if (sy.directive!=end) halt(2);
    if (listing_flag) fprintf(list_file,"%-12.12s:",
    module_name);
    i=0;
    for (sec=sec_liste; sec!=NULL; sec=sec->sec_next)
    {
       if (listing_flag)
       {
          if (i++>=3) { fprintf(list_file,
          "\n"); i=0; }
          fprintf(list_file," %8.8s=%08X",
                   sec->sec_name,sec->sec_length-sec->
                   sec_oldlen);
       }
       sec->sec_oldlen=sec->sec_length;
    }
    if (listing_flag) fprintf(list_file,"\n");
    strcpy(module_name,"NO MODULE");
    nxsy();
}

void      calc_xref(x,c,modname)
XREF      *x;
char      *c,*modname;
{
    short i;
    long  value;

    value = x->xref_abs;
    c += x->xref_pos;
    /* printf("XREF at %8X %X",c-membot,value); */
    for (i=0; i<x->xref_ops; i++)
    {
       /*printf("%c",x->xref_oper[i].xop_oper);*/
```



ATARI ST SOFTWARE
Regent Base 2.0
Regent Base ist eine relationale Datenbank, die über einen umfangreichen SQL Befehlssatz verfügt. SQL ist eine 4GL Sprache, die sich einer immer größer werdenden Beliebtheit bei professionellen Anwendern erfreut.
Ihre Daten lassen sich einfach durch die leicht zu erstellenden GEM Formulare bearbeiten. Regent Base kennt keine Grenze für die Größe Ihrer Datenbank. Sie können jederzeit Felder hinzufügen oder entfernen, wenn sich Ihre Bedürfnisse ändern. Datensätze können mit dBase kompatiblen Programmen ausgetauscht werden.
* 349,– DM.
the Informer
The Informer ist eine leicht zu bedienende Datenbank unter GEM, mit der Text und Bilder bearbeitet werden können. Alle Funktionen sind leicht über die Maus oder die Tastatur aufzurufen und machen eine Programmierung überflüssig. The Informer ist unglaublich schnell, da alle Daten im Speicher verfügbar sind. Die Daten können als GEM Formular oder (zur besseren Übersicht) in einer Tabelle dargestellt werden.
* 249,– DM.
Unverbindliche Preisempfehlung
Programmierer gesucht
Bei Ihrem Atari Händler oder direkt von:
COMPUTERWARE
Computerware Gerd Sender, Moselstraße 39, 5000 Köln 50, Telefon: 0221-392583

```
        switch(x->xref_oper[i].xop_optyp)
        {
           case -1 : if (x->xref_oper[i].xop_oper=='+')
                        value+=x->xref_oper[i].xop_ptr.
                        xop_sec->sec_start
                                  -membot;
                     else value-=x->xref_oper[i].xop_ptr.
                        xop_sec->sec_start
                                  -membot;
                     /*printf("%s/%x",x->xref_oper[i].
                     xop_ptr.xop_sec->sec_name,
                        x->xref_oper[i].xop_ptr.xop_sec
                        ->sec_start);*/
                     break;
           case  0 : if (x->xref_oper[i].xop_oper=='+')
                        value+=c-membot;
                     else value-=c-membot;
                     printf("&");
                     break;
           case  1 : if (x->xref_oper[i].xop_oper=='+')
                        value+=calc_xsy(x->xref_oper[i]
                        .xop_ptr.xop_sym);
                     else value-=calc_xsy(x->xref_oper[i]
                        .xop_ptr.xop_sym);
                     /*printf("%s/%x",x->xref_oper[i]
                     .xop_ptr.xop_sym->xsy_name,
                        calc_xsy(x->xref_oper[i].xop_ptr
                        .xop_sym));*/
                     break;
        }
        /*printf("\n");*/
     }
     if (c<membot || c>memstart) halt(2);
     if (x->xref_trunc & 32) value-=c-membot;
     switch( x->xref_trunc & 7 )
     {
        case 4 : if (((long)c)&1) halt(1); *((long*)c)=
                 value; break;
        case 2 : if (((long)c)&1) halt(1); *((short*)c)=
                 value;
                 if (x->xref_trunc & 8) /* Wert hat
                                           Vorzeichen */
                 {
                    if (value<-32768L || value>32767L)
                    {
                       range_err++;
                       printf("XREF.W-value out of range
                       in module '%s'\n",
                              modname);
                    }
                 }
                 else /* Value ist unsigned */
                 {
                    if (value>65535L)
                    {
                       range_err++;
                       printf("XREF.UW-value out of Range
                       in module '%s'\n",
                              modname);
                    }
                 }
                 break;
        case 1 : *c=value;
                 if (x->xref_trunc & 8) /* Wert hat
                                           Vorzeichen */
                 {
                    if (value<-128L || value>128L)
                    {
                       range_err++;
                       printf("XREF.B-value out of Range
                       in module '%s'\n",
                              modname);
                    }
                 }
                 else /* Value ist unsigned */
                 {
                    if (value>255L)
                    {
                       range_err++;
                       printf("XREF.UB-value out of Range
                       in module '%s'\n",
                              modname);
                    }
                 }
                 break;
        default: halt(2);
     }
     if ( x->xref_trunc & 64 )
     {
        if (altxref==NULL)
        {
           if (code_ptr>=memtop-4) halt(1);
           altxref=c;
           *((long*)code_ptr)=c-membot;
           code_ptr+=4;
        }
        else
        {
           while(c-altxref>254)
           {
              altxref+=254;
              if (code_ptr>=memtop) halt(1);
              *code_ptr++='\001';
           }
           if (code_ptr>=memtop) halt(1);
           *code_ptr++=c-altxref;
           altxref=c;
        }
     }
}

void     all_xrefs(sec_liste)
SECTION  *sec_liste;
{
   XREF  *x;

   while(sec_liste!=NULL)
   {
      x=(XREF*)sec_liste->sec_fxref;
      while(x!=NULL)
      {
         calc_xref(x,sec_liste->sec_start,
         sec_liste->sec_module->mod_name);
         x=x->xref_next;
      }
      sec_liste=sec_liste->sec_next;
   }
   if (altxref==NULL)
   {
      if (code_ptr>=memtop-8) halt(1);
      *((long*)code_ptr)=0; code_ptr+=4;
      *((long*)code_ptr)=0; code_ptr+=4;
   }
   else
   {
      if (code_ptr>=memtop) halt(1);
      *code_ptr++='\0';
   }
}

void  init_mem()
{
   pdef_name=(ALFA*)malloc(MAX_PDEF*sizeof(ALFA));
   ndef_name=(ALFA*)malloc(MAX_NDEF*sizeof(ALFA));
   membot=(char*)malloc(mem_size);
   module_buffer=(char*)malloc(buf_size);
   if ( membot==NULL     || module_buffer==NULL ||
        ndef_name==NULL || pdef_name==NULL )  halt(0);

   memtop=membot+mem_size;
   memstart=membot;
   memend=memtop;
   altstart=NULL;
   module_top=module_buffer+buf_size;
   module_max=module_ptr=module_end=module_buffer;

}

void  make_ext(make_name,name,ext)
char  *make_name,*name,*ext;
{
   char  oldext[FMSIZE];

   stcgfe(oldext,name);
   if (!*oldext) strmfe(make_name,name,ext);
   else strcpy(make_name,name);
}

void  command_line(argc,argv)
int   argc;
char  *argv[];
{
   int   i=2;
   int   x;

   if (argc<2) { printf("No Filename specified\n");
   exit(1); }
   strcpy(file_name,argv[1]);
```

```
    strmfe(input_name,file_name,"BIN");
    strmfe(listing_name,file_name,"MAP");
    strmfe(control_name,file_name,"LNK");
    strmfe(program_name,file_name,"PRG");
    listing_flag=0;
    control_flag=0;
    program_flag=1;
    debug_flag=0;
    symbol_flag=0;
    spar_flag=0;
    i=2;
    if (argc>i) if (*argv[i]!='-')
        { make_ext(control_name,argv[i++],"LNK");
        control_flag=1; }
    if (argc>i) if (*argv[i]!='-')
        { make_ext(listing_name,argv[i++],"MAP");
        listing_flag=1; }
    if (argc>i) if (*argv[i]!='-') make_ext
        (program_name,argv[i++],"PRG");
    for (;i<argc;i++)
    {
        if (!stricmp(argv[i],"-NOLIST"))
        { listing_flag=0; continue; }
        if (!stricmp(argv[i],"-NODEBUG"))
        { debug_flag=0;continue; }
        if (!stricmp(argv[i],"-NOPROG"))
        { program_flag=0; continue; }
        if (!stricmp(argv[i],"-DEBUG"))
        { debug_flag=1;continue; }
        if (!stricmp(argv[i],"-SYM"))
        { symbol_flag=1;  continue; }
        if (!stricmp(argv[i],"-NOSYM"))
        { symbol_flag=0;  continue; }
        if (!stricmp(argv[i],"-SEC"))
        { spar_flag=1;continue; }
        if (!stricmp(argv[i],"-NOSEC"))
        { spar_flag=0;continue; }
        if (!stricmp(argv[i],"-WITH"))
        {
            if (i+1<argc) if (*argv[i+1]!='-')
                strcpy(control_name,argv[++i]);
            control_flag=1;
            continue;
        }
        if (!stricmp(argv[i],"-LIST"))
        {
            if (i+1<argc) if (*argv[i+1]!='-')
                strcpy(listing_name,argv[++i]);
            listing_flag=1;
            continue;
        }
        if (!stricmp(argv[i],"-PROG"))
        {
            if (i+1<argc) if (*argv[i+1]!='-')
                strcpy(program_name,argv[++i]);
            program_flag=1;
            continue;
        }
        if (!stricmp(argv[i],"-MEM"))
        {
            if (i+1<argc) if (*argv[i+1]!='-')
            {
                x=atoi(argv[++i]);
                if (x>2 && x<800) mem_size=x*1024;
            }
            continue;
        }
        if (!stricmp(argv[i],"-BUF"))
        {
            if (i+1<argc) if (*argv[i+1]!='-')
            {
                x=atoi(argv[++i]);
                if (x>2 && x<800) buf_size=x*1024;
            }
            continue;
        }
        printf("Invalid Option:'%s'\n",argv[i]);
    }
}

int   test_module()
{
    register int c;
    int      end_test=0;
    int      result=0;
    register short    i;
    char     string[80];
    register char *p;
    XSYMBOL  *s;

    module_end=module_ptr=module_buffer;

    do
    {
        c=get_drct(); if (c<0) halt(2);
        if (module_end+128>=module_top) halt(9);
        *module_end++=c;
        if (c==0xFB)
        {
            c=get_drct(); *module_end++=c;
            switch(c)
            {
                case   -1 : halt(2); break;
                case  0xFB : break;
                case  0x01 :
                case  0x02 : i=get_drct(); *module_end++=i;
                             while(i–) *module_end++=
                             get_drct();
                             break;
                case  0x03 :
                case  0x05 : *module_end++=get_drct();
                             *module_end++=get_drct();
                             *module_end++=get_drct();
                             *module_end++=get_drct();
                             break;
                case  0x12 :
                case  0x04 : *module_end++=get_drct();
                             *module_end++=get_drct();
                             break;
                case  0x06 : p=string;
                             i=get_drct(); *module_end++=i;
                             while(i–)
                             {*p=get_drct(); *module_end++=
                             *p++; }
                             *p='\0';
                             *module_end++=get_drct();
                             *module_end++=get_drct();
                             *module_end++=get_drct();
                             *module_end++=get_drct();
                             *module_end++=get_drct();
                             *module_end++=get_drct();
                             s=src_xsy(xsy_liste,string);
                             if (s!=NULL)
                             if (!(s->xsy_defd&1))
                               end_test=result=1;
                             break;
                case  0x07 : *module_end++=get_drct();
                             *module_end++=get_drct();
                             *module_end++=get_drct();
                             *module_end++=get_drct();
                             *module_end++=get_drct();
                             while(1)
                             {
                               c=get_drct();
                               *module_end++=c;
                               if (c==0xFB || c==-1) break;
                               *module_end++=get_drct();
                               *module_end++=get_drct();
                             }
                             break;
                case  0x10 : *module_end++=get_drct();
                             *module_end++=get_drct();
                             i=get_drct(); *module_end++=i;
                             while(i–) *module_end++=
                             get_drct();
                             break;
                case  0x13 : end_test=1;
                             break;
                default      : halt(2);
            }
        } /* if 0xFB */
    } while(!end_test);
    if (module_end>module_max) module_max=module_end;
    return(result);
}

void  link_file(name,lib_mode)
char  *name;
int   lib_mode;
{
    inp_hnd=Fopen(name,0);
    buf_ptr=buf_end=inp_buf+BLEN;
    strcpy(module_name,"*NO MODULE");
    if (inp_hnd<0)
    {
        printf("Cannot open binary file: '%s'\n",name);
        exit(1);
    }
```

```
    nxsy(); /* SOURCE oder EOFSY */
    while(sy.directive!=eofsy)
    {
       if (lib_mode)
       {
          if (!test_module())
          {
             module_end=module_ptr=module_buffer;
             nxsy();
          }
          else module();
       }
       else module();
    }
    Fclose(inp_hnd);
}

void  write_prog()
{
   int       handle;
   int       n,h;
   SECTION *sec;
   char     *start,*endcode;

   if ( (handle=Fcreate(program_name,0))<0 )
   {
      printf("Cannot open %s for write\n"); halt(10);
   }
   *((long*)&header[7])=debug_end-memstart;
   n=Fwrite(handle,28,header); if (n!=28) halt(10);

   if (spar_flag)
   {
      h=1;
      sec=sec_liste;
      start=membot;

      while(sec->sec_id>-2 && sec->sec_next!=NULL) sec=
      sec->sec_next;
      if (sec->sec_id==-2) /* TEXT-Section schreiben */
      {
         endcode  = sec->sec_start;
         *((long*)&header[h])=endcode-start;
         if (endcode-start)
         {
            n=Fwrite(handle,endcode-start,start);
            if (n!=endcode-start) halt(10);
         }
         h=3; /* next SECTION */
         start=endcode; /* neuer Start für den Rest */
      }

      while(sec->sec_id>-3 && sec->sec_next!=NULL) sec=
      sec->sec_next;
      if (sec->sec_id==-3)
      {  /* BSS gefunden */
         endcode  = sec->sec_start;
         {
            *((long*)&header[h])=endcode-start;
            if (endcode-start)
            {
               n=Fwrite(handle,endcode-start,start);
               if (n!=endcode-start) halt(10);
            }
         }
         h=5; /* next SECTION */
         start=endcode; /* neuer Start für den Rest */
      }

      /* Rest in (h)-Section schreiben */
      endcode=debug_start;
      *((long*)&header[h])=endcode-start;
      if (h<5) /* BSS-Section wird nicht geschrieben */
      {
         if (endcode-start)
         {
            n=Fwrite(handle,endcode-start,start);
            if (n!=endcode-start) halt(10);
         }
      }
   }
   else /* also nicht sparen */
   {
      start=membot;
      endcode=debug_start;
      *((long*)&header[1])=debug_start-membot;
      if (endcode-start)
      {
         n=Fwrite(handle,endcode-start,start);
         if (n!=endcode-start) halt(10);
      }
   }
   /* Symboltabelle und Relocation Table schreiben */
   *((long*)&header[7])=debug_end-debug_start;
   start=debug_start;
   endcode=code_ptr;
   if (endcode-start)
   {
      n=Fwrite(handle,endcode-start,start);
      if (n!=endcode-start) halt(10);
   }
   Fseek(0,handle,0);
   n=Fwrite(handle,28,header); if (n!=28) halt(10);
   if (Fclose(handle)<0) halt(10);
}

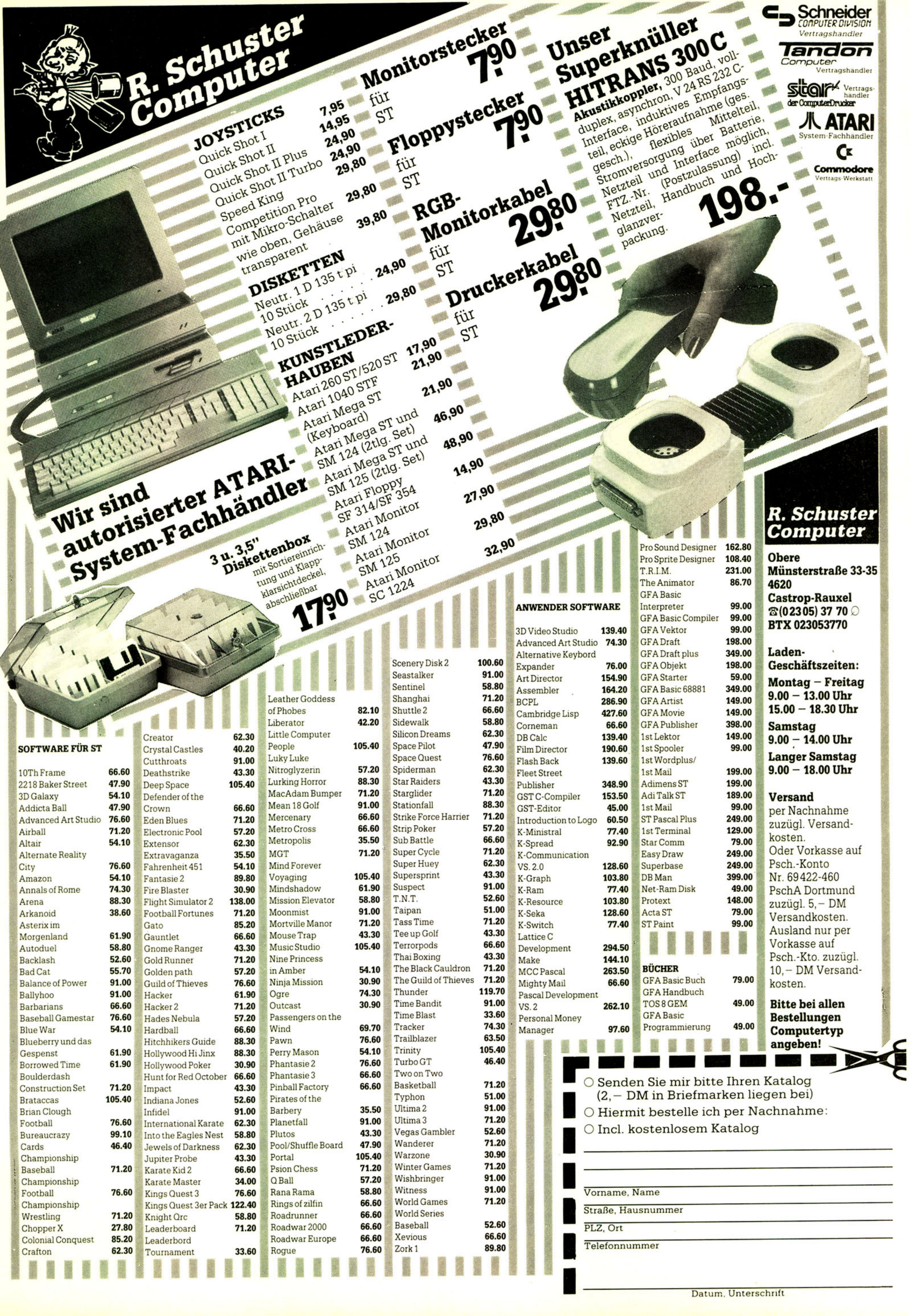
void  main(argc,argv)
int   argc;
char  *argv[];
{

   char  line[80],*name;
   FILE  *fp;
   int   lib_mode;
   int   err_code=0;

   printf(message);
   mem_size=100*1024; buf_size=32*1024;
   double_sym=undefd_sym=range_err=0;
   command_line(argc,argv);
   mod_liste=NULL;
   list_file=stdout;
   if (listing_flag)
   {
      list_file=fopen(listing_name,"w");
      if (list_file==NULL)
      {
         printf("Cannot open list-file '%s' for
         write\n"); exit(1);
      }
      if (list_file!=stdout) fprintf(list_file,message)
   }

   init_mem();
   sec_liste=curr_sec=moved_sec=NULL;
   xsy_liste=NULL;
   xref_liste=NULL;
   altxref=NULL;

   if (control_flag)
   {
      fp=fopen(control_name,"r");
      if (fp==NULL)
      {
         printf("Cannot open control-file:'%s'\n",
         control_name);
         exit(1);
      }
      while(!feof(fp))
      {
         if ((name=fgets(line,80,fp))==NULL) *line='\0'
         for (name=line; *name!='\n' && *name; name++);
         *name='\0';
         if (!*line) continue;
         if (*line=='*') continue;
         name=NULL;
         lib_mode=0;
         if (!strnicmp(line,"INPUT",5)) name=line+5;
         else if (!strnicmp(line,"LIBRARY",7))
         { name=line+7; lib_mode=1; }
         if (name==NULL) printf("Invalid control-line:
         %s\n",line);
         else
         {
            name=stpblk(name);
            if (*name=='*') strmfe(name,file_name,
            "BIN");
            else make_ext(name,name,"BIN");
            link_file(name,lib_mode);
         }
      }
      fclose(fp);
   }
   else link_file(input_name,0);
   code_ptr=memstart;
   debug_start=code_ptr;
   if (debug_flag) debug_table(xsy_liste);
```


R. Schuster Computer
JOYSTICKS
Quick Shot I 7,95
Quick Shot II 14,95
Quick Shot II Plus 24,90
Quick Shot II Turbo 24,90
Speed King 29,80
Competition Pro mit Mikro-Schalter 29,80
wie oben, Gehäuse transparent 39,80
DISKETTEN
Neutr. 1 D 135 t pi 10 Stück 24,90
Neutr. 2 D 135 t pi 10 Stück 29,80
KUNSTLEDER-HAUBEN
Atari 260 ST/520 ST 17,90
Atari 1040 STF 21,90
Atari Mega ST (Keyboard) 21,90
Atari Mega ST und SM 124 (2tlg. Set) 46,90
Atari Mega ST und SM 125 (2tlg. Set) 48,90
Atari Floppy SF 314/SF 354 14,90
Atari Monitor SM 124 27,90
Atari Monitor SM 125 29,80
Atari Monitor SC 1224 32,90
Monitorstecker 7.90 für ST
Floppystecker 7.90 für ST
RGB-Monitorkabel 29.80 für ST
Druckerkabel 29.80 für ST
Unser Superknüller HITRANS 300 C
Akustikkoppler, 300 Baud, vollduplex, asynchron, V 24 RS 232 C-Interface, induktives Empfangsteil, eckige Höreraufnahme (ges. gesch.), flexibles Mittelteil, Stromversorgung über Batterie, Netzteil und Interface möglich, FTZ.-Nr. (Postzulassung) incl. Netzteil, Handbuch und Hochglanzverpackung.
198.-
Schneider COMPUTER DIVISION Vertragshandler
Tandon Computer Vertragshandler
star Vertragshändler der ComputerDrucker
ATARI System-Fachhändler
Commodore Vertrags-Werkstatt
Wir sind autorisierter ATARI-System-Fachhändler
3 u. 3,5" Diskettenbox
mit Sortiereinrichtung und Klappklarsichtdeckel, abschließbar
17.90
SOFTWARE FÜR ST
10Th Frame 66.60
2218 Baker Street 47.90
3D Galaxy 54.10
Addicta Ball 47.90
Advanced Art Studio 76.60
Airball 71.20
Altair 54.10
Alternate Reality City 76.60
Amazon 54.10
Annals of Rome 74.30
Arena 88.30
Arkanoid 38.60
Asterix im Morgenland 61.90
Autoduel 58.80
Backlash 52.60
Bad Cat 55.70
Balance of Power 91.00
Ballyhoo 91.00
Barbarians 66.60
Baseball Gamestar 76.60
Blue War 54.10
Blueberry und das Gespenst 61.90
Borrowed Time 61.90
Boulderdash Construction Set 71.20
Brataccas 105.40
Brian Clough Football 76.60
Bureaucrazy 99.10
Cards 46.40
Championship Baseball 71.20
Championship Football 76.60
Championship Wrestling 71.20
Chopper X 27.80
Colonial Conquest 85.20
Crafton 62.30
Creator 62.30
Crystal Castles 40.20
Cutthroats 91.00
Deathstrike 43.30
Deep Space 105.40
Defender of the Crown 66.60
Eden Blues 71.20
Electronic Pool 57.20
Extensor 62.30
Extravaganza 35.50
Fahrenheit 451 54.10
Fantasie 2 89.80
Fire Blaster 30.90
Flight Simulator 2 138.00
Football Fortunes 71.20
Gato 85.20
Gauntlet 66.60
Gnome Ranger 43.30
Gold Runner 71.20
Golden path 57.20
Guild of Thieves 76.60
Hacker 61.90
Hacker 2 71.20
Hades Nebula 57.20
Hardball 66.60
Hitchhikers Guide 88.30
Hollywood Hi Jinx 88.30
Hollywood Poker 30.90
Hunt for Red October 66.60
Impact 43.30
Indiana Jones 52.60
Infidel 91.00
International Karate 62.30
Into the Eagles Nest 58.80
Jewels of Darkness 62.30
Jupiter Probe 43.30
Karate Kid 2 66.60
Karate Master 34.00
Kings Quest 3 76.60
Kings Quest 3er Pack 122.40
Knight Orc 58.80
Leaderboard 71.20
Leaderbord Tournament 33.60
Leather Goddess of Phobes 82.10
Liberator 42.20
Little Computer People 105.40
Luky Luke
Nitroglyzerin 57.20
Lurking Horror 88.30
MacAdam Bumper 71.20
Mean 18 Golf 91.00
Mercenary 66.60
Metro Cross 66.60
Metropolis 35.50
MGT 71.20
Mind Forever Voyaging 105.40
Mindshadow 61.90
Mission Elevator 58.80
Moonmist 91.00
Mortville Manor 71.20
Mouse Trap 43.30
Music Studio 105.40
Nine Princess in Amber 54.10
Ninja Mission 30.90
Ogre 74.30
Outcast 30.90
Passengers on the Wind 69.70
Pawn 76.60
Perry Mason 54.10
Phantasie 2 76.60
Phantasie 3 66.60
Pinball Factory 66.60
Pirates of the Barbery 35.50
Planetfall 91.00
Plutos 43.30
Pool/Shuffle Board 47.90
Portal 105.40
Psion Chess 71.20
Q Ball 57.20
Rana Rama 58.80
Rings of zilfin 66.60
Roadrunner 66.60
Roadwar 2000 66.60
Roadwar Europe 66.60
Rogue 76.60
Scenery Disk 2 100.60
Seastalker 91.00
Sentinel 58.80
Shanghai 71.20
Shuttle 2 66.60
Sidewalk 58.80
Silicon Dreams 62.30
Space Pilot 47.90
Space Quest 76.60
Spiderman 62.30
Star Raiders 43.30
Starglider 71.20
Stationfall 88.30
Strike Force Harrier 71.20
Strip Poker 57.20
Sub Battle 66.60
Super Cycle 71.20
Super Huey 62.30
Supersprint 43.30
Suspect 91.00
T.N.T. 52.60
Taipan 51.00
Tass Time 71.20
Tee up Golf 43.30
Terrorpods 66.60
Thai Boxing 43.30
The Black Cauldron 71.20
The Guild of Thieves 71.20
Thunder 119.70
Time Bandit 91.00
Time Blast 33.60
Tracker 74.30
Trailblazer 63.50
Trinity 105.40
Turbo GT 46.40
Two on Two Basketball 71.20
Typhon 51.00
Ultima 2 91.00
Ultima 3 71.20
Vegas Gambler 52.60
Wanderer 71.20
Warzone 30.90
Winter Games 71.20
Wishbringer 91.00
Witness 91.00
World Games 71.20
World Series Baseball 52.60
Xevious 66.60
Zork 1 89.80
ANWENDER SOFTWARE
3D Video Studio 139.40
Advanced Art Studio 74.30
Alternative Keybord Expander 76.00
Art Director 154.90
Assembler 164.20
BCPL 286.90
Cambridge Lisp 427.60
Corneman 66.60
DB Calc 139.40
Film Director 190.60
Flash Back 139.60
Fleet Street Publisher 348.90
GST C-Compiler 153.50
GST-Editor 45.00
Introduction to Logo 60.50
K-Ministral 77.40
K-Spread 92.90
K-Communication VS. 2.0 128.60
K-Graph 103.80
K-Ram 77.40
K-Resource 103.80
K-Seka 128.60
K-Switch 77.40
Lattice C Development 294.50
Make 144.10
MCC Pascal 263.50
Mighty Mail 66.60
Pascal Development VS. 2 262.10
Personal Money Manager 97.60
Pro Sound Designer 162.80
Pro Sprite Designer 108.40
T.R.I.M. 231.00
The Animator 86.70
GFA Basic Interpreter 99.00
GFA Basic Compiler 99.00
GFA Vektor 99.00
GFA Draft 198.00
GFA Draft plus 349.00
GFA Objekt 198.00
GFA Starter 59.00
GFA Basic 68881 349.00
GFA Artist 149.00
GFA Movie 149.00
GFA Publisher 398.00
1st Lektor 149.00
1st Spooler 99.00
1st Wordplus/1st Mail 199.00
Adimens ST 199.00
Adi Talk ST 189.00
1st Mail 99.00
ST Pascal Plus 249.00
1st Terminal 129.00
Star Comm 79.00
Easy Draw 249.00
Superbase 249.00
DB Man 399.00
Net-Ram Disk 49.00
Protext 148.00
Acta ST 79.00
ST Paint 99.00
BÜCHER
GFA Basic Buch 79.00
GFA Handbuch TOS 8 GEM 49.00
GFA Basic Programmierung 49.00
R. Schuster Computer
Obere Münsterstraße 33-35
4620 Castrop-Rauxel
(02305) 37 70
BTX 023053770
Laden-Geschäftszeiten:
Montag – Freitag
9.00 – 13.00 Uhr
15.00 – 18.30 Uhr
Samstag
9.00 – 14.00 Uhr
Langer Samstag
9.00 – 18.00 Uhr
Versand
per Nachnahme zuzügl. Versandkosten. Oder Vorkasse auf Psch.-Konto Nr. 69422-460 PschA Dortmund zuzügl. 5,– DM Versandkosten. Ausland nur per Vorkasse auf Psch.-Kto. zuzügl. 10,– DM Versandkosten.
Bitte bei allen Bestellungen Computertyp angeben!
○ Senden Sie mir bitte Ihren Katalog (2,– DM in Briefmarken liegen bei)
○ Hiermit bestelle ich per Nachnahme:
○ Incl. kostenlosem Katalog
Vorname, Name
Straße, Hausnummer
PLZ, Ort
Telefonnummer
Datum, Unterschrift

```
    debug_end=code_ptr;
    all_xrefs(sec_liste);

    statistic();
    fprintf(list_file,"Program length     = %8X\n",
    memstart-membot);
    fprintf(list_file,"Symbol Table       = %8X\n",
    debug_end-memstart);
    fprintf(list_file,"Relocation Table = %8X\n",
    code_ptr-debug_end);
    fprintf(list_file,"-------------------\n");
    fprintf(list_file,
       "Memory Usage       = %7d%%\n",
       (code_ptr-membot)*100/mem_size);
    fprintf(list_file,
       "Buffer Usage       = %7d%%\n",
       (module_max-module_buffer)*100/buf_size);
    fprintf(list_file,"-------------------\n");
    list_xsy(xsy_liste,1);
    if (program_flag) write_prog();
    if (symbol_flag)
    {
       fprintf(list_file,"\nSymbol Table:\n");
       fprintf(list_file,"---------\n");
       list_xsy(xsy_liste,0);
       fprintf(list_file,"\n");
    }
    if (undefd_sym)
    {
       if (list_file!=stdout)
          fprintf(list_file,"Undefined Symbols:%8d\n",
          undefd_sym);
       printf("Undefined Symbols:%8d\n",undefd_sym);
       err_code=1;
    }
    if (double_sym)
    {
       if (list_file!=stdout)
          fprintf(list_file,"Multiply defined :%8d\n",
          double_sym);
       printf("Multiply defined :%8d\n",double_sym);
       err_code=1;
    }
    if (range_err)
    {
       if (list_file!=stdout)
          fprintf(list_file,"Range errors      :%8d\n",
          range_err);
       printf("Range errors      :%8d\n",range_err);
       err_code=1;
    }

    printf("\nLink completed\n");
    if (list_file!=stdout) fprintf(list_file,
    "\nLink completed\n");
    if (listing_flag) fclose(fp);
    exit(err_code);
}
```

NICHT´S DABEI GEDACHT?

Das neue (alte) Wordplus

TEST

Eine neue Version von Wordplus ist auf dem Markt. Die Versionsnummer 2.02 scheint umfassende Neuerungen gegenüber der alten Version 1.89 anzudeuten. Das Wichtigste in Kürze.

1. Wordplus ist in den Grundfunktionen nicht umfassend verändert worden. Die gewohnten Menüs sind bis auf das Untermenü »Alles beenden« gleichgeblieben. Hier gibt es die Möglichkeit, auch bei vier geöffneten Fenstern sofort das Textprogramm zu verlassen. Nach wie vor ist Wordplus beim Laden von Dateien, Scrollen im Text und Suchen/Ersetzen entsetzlich langsam und träge. Offensichtliche Mängel der alten Version hat man nicht behoben: die Trennhilfe arbeitet immer noch nicht bei in Anführungszeichen eingeschlossenen Wörtern, die Statistikfunktion suggeriert mehr freien Speicherplatz im RAM als tatsächlich vorhanden ist, und es fehlt ein Fußnotenziffern-Offset für das Verketten von längeren Texten mit Anmerkungen. Ob die neue Version von Wordplus ebenfalls längere Texte »zerstückelt«, konnte bisher noch nicht festgestellt werden.

2. Grundlegend neugestaltet wurde hingegen die Rechtschreibüberprüfung. Das neue Konzept läßt sich wie folgt umschreiben: ein mitgeliefertes Hauptwörterbuch enthält die wichtigsten Wörter in komprimierter Datenspeicherung. Dieses Wörterbuch kann man erweitern, aber nicht kürzen. Wählt der Anwender »Wort hinzufügen« an, wird das neue Wort zunächst im RAM gespeichert und bei Arbeitsende als Zusatzwörterbuch auf Diskette ausgelagert. Beim nächsten Start lädt man das persönliche Lexikon, bei dem es sich um eine alphabetisch sortierte ASCII-Liste handelt, wieder mit. Allerdings kann das Zusatzwörterbuch nur direkt nach dem Hauptwörterbuch geladen werden: später nicht mehr. Mit Hilfe des Programms »Dicmerge« läßt sich eine ASCII-Datei an das Hauptwörterbuch anhängen. Es empfiehlt sich allerdings, mit dieser Option vorsichtig umzugehen. Wenn nämlich ein falsches Wort ins Hauptwörterbuch aufgenommen wurde, läßt es sich von dort nicht mehr entfernen!

3. Programmtechnisch läuft das Laden des Wörterbuches und einer Zusatzdatei äußerst langsam ab. Nach dem eigentlichen Ladevorgang erscheint die Meldung »Tabellen in Bau - bitte warten Sie«. Bis zu zweieinhalb Minuten (!) braucht Word-

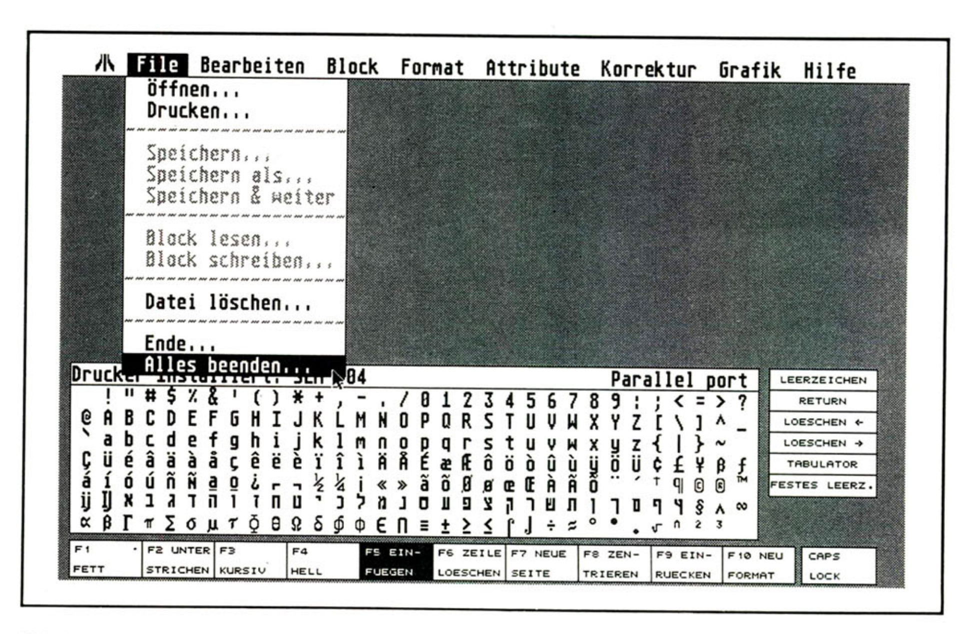
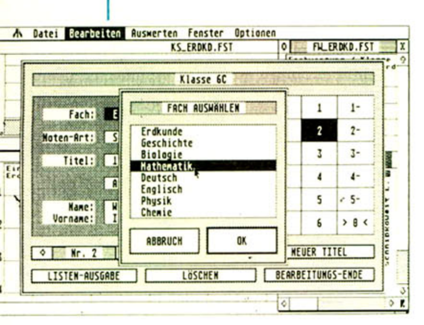
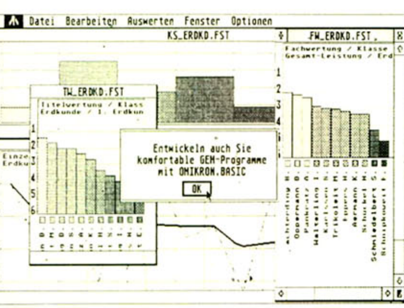


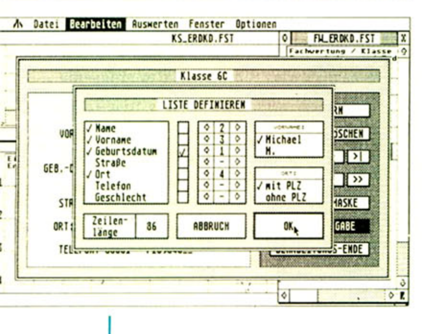
Bild 1: Ein neuer Menüpunkt.:"Alles beenden"

OMIKRON.BASIC: „Sieger aller Klassen"

(68000er Magazin, Oktober 1987, Seite 57)







eine gute Nachricht für alle GFA-Basic-Benutzer: OSKAR* wandelt Ihre Programme in OMIKRON.BASIC-Programme um.

„Der OMIKRON.Basic-Interpreter stellt zusammen mit dem Compiler das beste Basic-Entwicklungssystem für den ATARI ST dar." (ST-Computer 9/87, Seite 65)

* OSKAR ist ein Programm der Markt & Technik Verlag AG. Sie finden es im Happy-Computer-Sonderheft „St-Magazin", Nr. 27.

ARITHMETIK: Rechengenauigkeit bis 19 Stellen bei allen Funktionen ● Rechenbereich bis 5.11 E±4931

GEM: Komplette GEM-Library ● Sämtliche AES- und VDI-Funktionen direkt mit Namen verfügbar ● Eigene BITBLIT-Routine

EXTRAS: Masken-INPUT ● SORT-Befehl sortiert beliebige Felder, auch mit Umlauten ● Matrizenbefehle

STRUKTUR: Prozeduren und mehrzeilige Funktionen mit Übergabe- und Rückgabe-Parametern und lokalen Variablen ● REPEAT... UNTIL, WHILE... WEND, mehrzeiliges IF... THEN... ELSE... ENDIF ● Labels bei GOTO, GOSUB, ON GOTO und ON GOSUB

EDITOR: Mit oder ohne Zeilennummern (umschaltbar) ● Drei Schriftgrößen bis 57 x 128 Zeichen ● Frei definierbare Teach-In-Funktionstasten

GESCHWINDIGKEIT: FIT-Code (FIT = Fast Interpreting Technique) ● Wir kennen keine schnelleren 68000-Fließkomma-Routinen ● Volle Integer-Arithmetik ● Eigene Disk-Routinen für beschleunigten Dateizugriff

KOMPATIBILITÄT: Für alle ATARI ST (auch MEGA ST) ● 99% MBASIC-kompatibel ● Editor findet Inkompatibilitäten, dadurch einfachste Anpassung

● Diskettenversion mit Demodiskette und Handbuch **DM 179,–**

● Modulversion mit Demodiskette und Handbuch **DM 229,–**

COMPILER

Noch mehr Tempo erreichen Ihre mit OMIKRON.BASIC entwickelten Applikationen durch unseren OMIKRON.BASIC-COMPILER.

Er erlaubt es Ihnen, bequem in Basic interpretativ zu programmieren, um nach der Compilierung Geschwindigkeiten zu erreichen, die bisher C-Compilern vorbehalten waren. Der Aufbruch in neue Dimensionen der Basic-Programmierung hat begonnen.

● Diskette mit Anleitung **DM 179,–**





... denn das Beste ist für Ihren ATARI ST gerade gut genug!

OMIKRON.SOFTWARE

Erlachstraße 15 · D-7534 Birkenfeld · ☎ (0 70 82) 53 86

Frankreich: OMIKRON · France, 11 Rue Dérodé, F-51100 Reims
Luxemburg: ELECOMP, 11 Avenue de la gare, L-4131 Esch-Alzette
Niederlande: Terminal Software Publicaties, Postbus 111, NL-5110 Baarle-Nassau
Österreich: Ueberreuter Media, Laudengasse 29, A-1082 Wien
Schweiz: Thali AG, Industriestraße 6, CH-6285 Hitzkirch

plus, um sein Lexikon im Speicher zu »entkomprimieren«. Nach der Ladeprozedur steht ein neuer Menüpunkt »Ständige Korrektur« zur Verfügung. Wenn dieser aktiviert ist, ertönt ein Signal bei jedem falsch geschriebenen Wort. Auch wenn die Arbeitsgeschwindigkeit ein bißchen darunter leidet, ist der »Online Spelling Checker« sicherlich eine gute Idee, - vor allem für »Blindschreiber«.

4. Aus dem mitgelieferten Hauptwörterbuch (176 KByte) können keine Worte entfernt werden. Das Lexikon bleibt so umfangreich, wie es ist. Als Ballast schleppt der Anwender also ständig auch die Wörter mit, von denen er mit Sicherheit weiß, daß er sie nie brauchen wird. Wir haben im deutschen Wörterbuch so seltene Worte wie »Sulz« (süddeutsch für »Sülze«) oder »Aare« (ein Fluß in der Schweiz) gefunden, während wichtige Begriffe wie »durchgängig«, »insofern«, »desweiteren«, »zwingend« oder »Schutz« fehlen. Schließlich gibt es dort nicht wenige Begriffe aus der Umgangssprache: allein vier Kombinationen mit einem Wort, das wir nicht drucken (»Sch...«), oder: »Sexbombe«, »Kauderwelsch«, »hauruck« und »mausetot«. Wie gesagt, dieses Stammwörterbuch nimmt allein rd. 200 KByte im Speicher ein, beansprucht eine lange Ladezeit und kann nicht gekürzt werden. Alle Versuche, ein leeres Lexikon zu erstellen, das man peu à peu selbst auffüllen würde, scheiterten. Wie erzeugt man etwa ein spanisches oder französisches Wörterbuch?

5. Ein Schlag ins Gesicht aller Anwender, die schon mit Version 1.89 gearbeitet haben, ist die Inkompatibilität der Lexika von Version 1.89 und 2.02. Wer mit viel Mühe ein eigenes Lexikon unter der alten Version erstellt hatte, kann damit jetzt nichts mehr anfangen. Wenn man schon ein neues Aufzeichnungsformat verwendet, wäre es ein Gebot des Anstands und der Fairness, für alle betroffenen Wordplus-Benutzer ein Konvertierungsprogramm beizugeben, das aus dem alten Lexikon ein neues erstellt!

6. Zusammenfassend sind es sechs Punkte, die GST und ATARI schleunigst umsetzen sollten, um zu retten, was noch zu retten ist:

- Entfernen von Wörtern auch aus dem Hauptlexikon,
- Zusatzwörterbücher, die man jederzeit laden kann,
- ein »leeres« Lexikon für den eigenen Bedarf,
- Tastaturbefehle für die wichtigsten Kommandos,
- Zusammenarbeit von Trennhilfe und Rechtschreibkorrektur,
- und last not least ein Konvertierungsprogramm für alte Lexika.

Meine Meinung: Die Programmierer von GST haben ein Jahr damit verbracht, ein fragwürdiges Lexikonkonzept zu entwickeln. Das Programm ist um keinen Deut schneller oder besser geworden. Anwenderwünsche wurden schlichtweg ignoriert. In diesem Jahr erscheinen WordPerfect, StarWriter und Tempus-Word für den Atari ST. Hoffen wir, daß eines der drei Programme den Nachfolger für Wordplus abgibt.

Michael Spehr

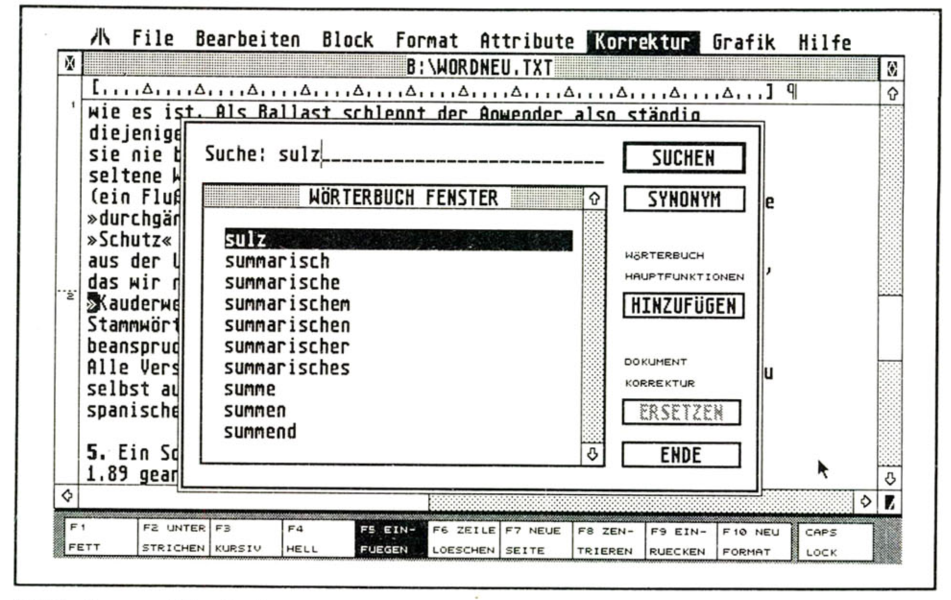


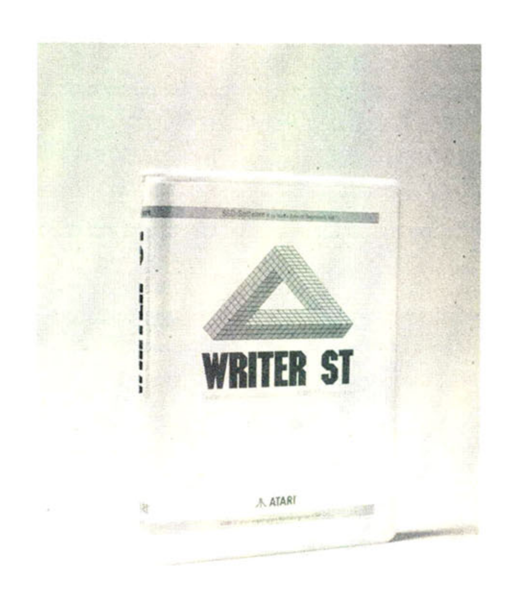
Bild 2: Das neue Wörterbuch

ENDE

SCHREIBEN MIT DEM WRITER ST

TEST

Mit WRITER ST gibt es wieder einmal ein neues Textverarbeitungsprogramm für den ST. Was kann es den bereits etablierten Textverarbeitungsprogrammen entgegensetzen, um auf dem Markt mitzuhalten?



Zielgruppen

Im Handbuch wird ausdrücklich darauf hingewiesen, daß WRITER ST für Benutzer gedacht ist, die viele Briefe oder kurze Dokumente, aber keine größeren Dokumentationen erstellen möchten. Ein WRITER ST-Dokument besteht aus maximal 10 Seiten.

Erster Eindruck

Beim ersten Aufruf des Programms wird zunächst eine Datei mit einer Kurzbeschreibung über WRITER ST geladen (Abb.1). Beim Aufbau des Bildschirms fällt auf, daß zwar die GEM-Menüleiste verwendet wird, das gewohnte Window allerdings durch ein einfaches Textfenster ohne Scroll-Balken ersetzt wird. Dieser gewonnene Platz wird für eine Darstellung der aktuellen Funktionstastenbelegung und ein Menü mit Seitenzahlen zum schnelleren Blättern in den (maximal 10!) Seiten am rechten Rand und ein Fenster zur Makrodefinition am unteren Rand genutzt. Sehr interessant ist die Idee, das Layout der kompletten aktuellen Seite in der rechten unteren Ecke des Bildschirms einzublenden. Leider ist dieses Bildchen sehr klein und kann nicht vergrößert dargestellt werden, so daß das Seitenlayout nur ungefähr zu erkennen ist.

Leistungsumfang

Die erste Schwierigkeit mit WRITER ST trat beim Versuch auf, eine neue Datei zu erstellen. Der Versuch, in der Dialogbox zum Laden eines Textes (wie üblich) einen neuen Dateinamen einzugeben, endete in der Fehlermeldung "Lesefehler! Text kann nicht gelesen werden!". Diese Fehlermeldung ist leider im Handbuch nicht dokumentiert (andere sind es). Abhilfe gibt es z.B. durch Speichern einer leeren Seite unter dem Namen des neuen Dokuments, das dann bearbeitet werden kann. WRITER ST kann auch ASCII-Dateien laden und Dokumente im ASCII-Format abspeichern.

Dokumente sind in WRITER ST seitenweise organisiert. Dadurch können einzelne Seiten eines Dokuments bearbeitet werden, ohne daß die übrigen Seiten verändert werden. Das bedeutet, daß beim Einfügen einer zusätzlichen Zeile in eine bereits volle Seite die letzte Zeile dieser Seite nicht auf die Folgeseite verschoben wird; die letzte Zeile der aktuellen Seite wird gelöscht oder die neue Zeile nicht eingefügt. Durch eine Option können Seiten auch als zusammenhängend deklariert werden, so daß auch größere Texte über mehrere Seiten hinweg eingegeben werden können. Auch Tabulatoren und rechte und linke Randbegrenzung beziehen sich immer auf eine Seite. Eine Seite besteht aus maximal 72 Zeilen und 70 Zeichen pro Zeile.

Will man mehrere (kurze!) Dokumente gleichzeitig bearbeiten, kann man diese seitenweise Organisation ausnutzen und ein Dokument an den Anfang des Arbeitsbereiches (z.B. Seiten 1 und 2) laden; das zweite Dokument kann dann ab Seite 3 eingeladen werden - sinnvollerweise werden die Seiten 2 und 3 in diesem Fall als nicht zusammenhängend deklariert, damit Text nicht von einem Dokument in das nächste verschoben werden kann (und damit das zweite Dokument versehentlich verändert würde). Verschiedene Windows für verschiedenen Dokumente können nicht angelegt werden.

Bei der **Bedienung** von WRITER ST fällt zunächst positiv auf, daß neben der Menüauswahl Kommandos auch über die Tastatur (CTRL-[Taste]) eingegeben werden können, so daß besonders geübte Benutzer relativ schnell arbeiten können. Einige Funktionen lassen sich allerdings nur in der Verbindung SHIFT-[linke Maustaste] ausführen (warum nicht nur über eine Maustaste?). Ungewöhnlich ist auch das Scrollen des Textausschnitts durch Bewegen der Maus über den oberen bzw. unteren Rand des Textausschnitts hinaus - Der Text wird allerdings sehr schnell bewegt.

geht weiter...

Angenehm ist es, daß man sich eine eigene Voreinstellung der Arbeitsumgebung abspeichern kann, dazu gehören Parameter wie Blocksatz, automatische Trennung, Sprungmarke in Formularmodus (s.u.) und auch zusätzliche Tastaturbelegungen (s.u.) und Seiten aus bis zu vier Dokumenten, die beim Starten von WRITER ST automatisch auf verschiedene Seiten des zu bearbeitenden Dokuments geladen werden. Bei der maximalen Seitenzahl von zehn Seiten ist das wohl nur für Briefe oder für ein einzelnes Dokument sinnvoll und im letzten Fall recht praktisch.

Auf der Verpackung wird darauf hingewiesen, daß mehrspaltige Texteingabe möglich ist. Im Handbuch sucht man allerdings vergeblich nach einer Beschreibung, wie das funktioniert, also ausprobieren! In dem am oberen Rand des Textausschnitts eingeblendeten Zeilenlineal stellt man zunächst die Randbegrenzungen für die erste Textspalte ein und gibt dann den Text ein, der in dieser ersten Spalte stehen soll. Am unteren Seitenrand angekommen, stellt man die Randbegrenzungen auf die zweite Textspalte ein, um anschließend den Text für diese Spalte einzugeben. Während die Randbegrenzung für die zweite Textspalte eingestellt ist, kann man nur noch diese Spalte edieren; es ist nicht möglich, den Cursor auf eine Stelle zu positionieren, die außerhalb der Randbegrenzungen liegt. Will man also nachträglich noch Änderungen an der ersten Textspalte vornehmen, muß die Randbegrenzung wieder verstellt werden (dann kann man aber den Cursor nicht mehr in die zweite Spalte bringen). Nachteilig ist auch, daß die Textspalten nicht logisch zusammenhängen, daß also Text am Ende der ersten

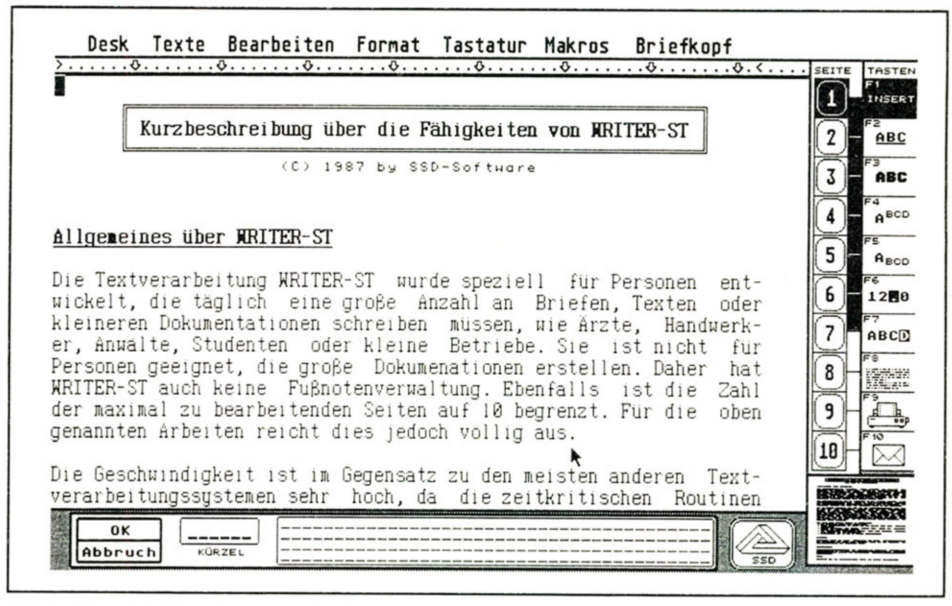


Bild 1: Einschaltbild mit Kurzbeschreibung

Spalte nicht automatisch in die zweite Textspalte verschoben wird, sondern auf die nächste Seite, (falls die Seiten als zusammenhängend deklariert sind, ansonsten wird er abgeschnitten).

WRITER ST arbeitet mit dem WYSIWYG-Prinzip (**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et), d.h. es kann auch Proportionalschrift dargestellt werden. Zu dem Programm werden verschiedene Fonts mitgeliefert, einem Dokument kann allerdings nur ein Font zugeordnet werden.

Es besteht die Möglichkeit, Textblöcke zu markieren (per Menüauswahl oder SHIFT-[Maustaste]), und diese Textblöcke zu verschieben, kopieren oder löschen. Eine Besonderheit besteht darin, daß auch beliebige mehrzeilige (rechteckige) Blöcke aus der Textmitte markiert werden können, die nicht über die gesamte Zeilenbreite gehen. Das kann recht angenehm sein, wenn man z.B. eine Art Tabelle erstellen möchte, man kann dann die Tabelleneinträge "spaltenweise" kopieren, verschieben oder löschen. Es ist aber nicht möglich, Blöcke zu definieren, die nicht über einen rechteckigen Bereich gehen (also z.B. in einer Zeilenmitte beginnen und dann über mehrere volle Zeilen reichen).

Eine Suchen/Ersetzen-Funktion ist auch vorhanden, allerdings nur mit Suchrichtung "vorwärts" und innerhalb eines markierten Blocks.

Der Zeilenumbruch bei der Texteingabe erfolgt halbautomatisch, d.h. der Randausgleich wird direkt beim Schreiben durchgeführt, Silbentrennung erfolgt aber erst beim Neu-formatieren eines Textblocks. Der Benutzer muß über eine Dialogbox jeden Trennvorschlag bestätigen - in diesem Fall ist das auch gut, da die meisten Trennvorschläge korrigiert werden müssen. Etwas umständlich ist es auch, nachträglich Text in einen Abschnitt einzufügen (Abb.2), da die restlichen Worte der zu verändernden Zeile jedes für sich in eine neue Zeile geschoben werden - bei einem Textwindow über nur 20 Zeilen kann ein Absatz dann schnell unübersichtlich werden - nach dem Formatieren wird natürlich die Zeile wieder zusammengefügt. Es ist nicht möglich, einen Textblock in die Mitte einer Zeile einzufügen, sondern nur zwischen zwei Zeilen oder im Modus "überschreiben".

Es gibt aber einige interessante Zusatzfunktionen bei WRITER ST. - Zum Beispiel die Möglichkeit, eigene Formulare zu erstellen (Abb.3). Ein Formular ist bei WRITER ST ein Dokument, bei dem nur noch gewisse Felder ausgefüllt und ediert werden können, der übrige Text aber bereits bei der Definition eines Formulars

geht weiter...

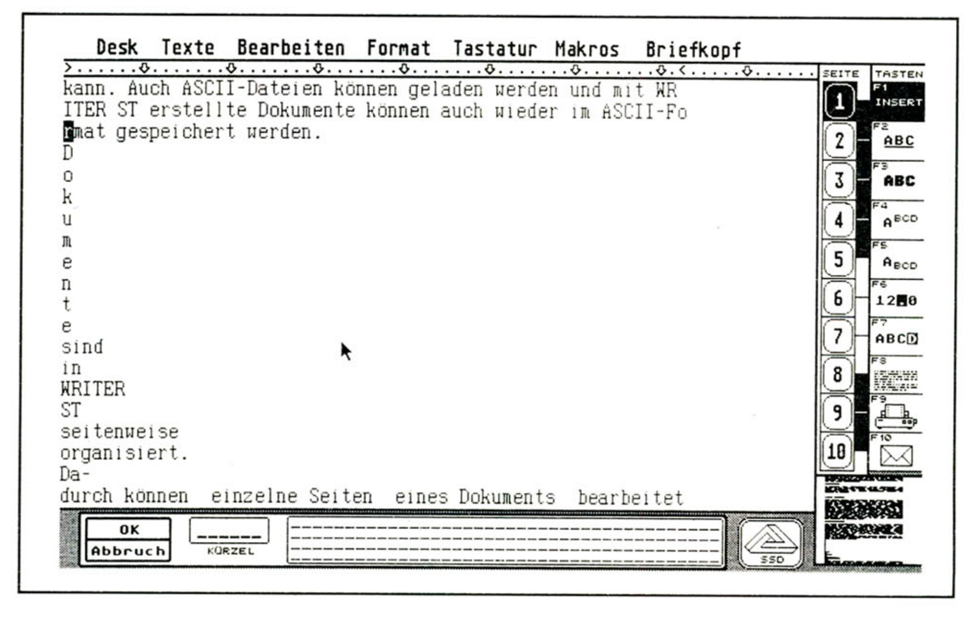


Bild 2: Nachträgliches Einfügen eines Textes

vorgegeben wird und beim Ausfüllen nicht mehr verändert werden kann. Für den Benutzer gibt es zwei Wege, eine Formularmaske zu definieren, die spätere Benutzung des fertigen Formulars ist in beiden Fällen gleich. Eine Möglichkeit besteht darin, ein beliebiges Zeichen der Tastatur (voreingestelllt ist ">") als Sprungmarke zu definieren; benutzt man ein Dokument als Formular (der Modus kann in dem Menü "Format" eingestellt werden - Formular ein/aus), springt der Cursor bei jedem Betätigen der RETURN-Taste zu der nächsten freien Sprungmarke. Die zweite Möglichkeit, ein Formular zu definieren, besteht darin, einen Block oder eine ganze Seite als Formular zu kennzeichnen (Menü Bearbeiten). In dem selektierten Bereich können nun einzelne Bereiche hervorgehoben werden, die später bei Benutzung des Formulars als Eingabefelder dienen.

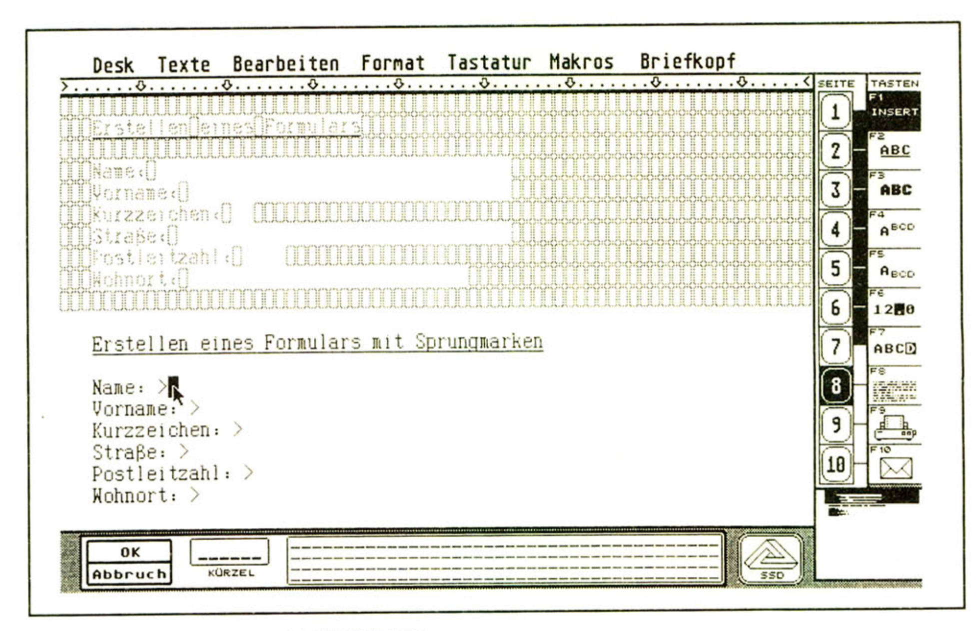


Bild 3: Formularerstellung mit WRITER ST

Als weitere Besonderheit bietet WRITER ST die Möglichkeit an, einen eigenen Briefkopf zu definieren (Abb.4). Ein solcher Briefkopf kann jeder beliebigen Seite eines Dokuments zugeordnet werden. Auf dem Bildschirm ist das an einer grauen Schattierung des dafür benötigten Platzes erkennbar. Eventuell an dieser Stelle im Dokument stehender Text wird nicht mit ausgedruckt (er würde sich ja sonst mit dem Briefkopf überschneiden), der Platz kann daher für Notizen benutzt werden und wird bei Ausblendung des Briefkopfs sichtbar.

Ein Briefkopf besteht aus einer Kopfzeile für den Namen des Absenders, einem "Briefkopf" zur näheren Erläuterung des Absenders, einer Absenderzeile für Briefumschläge mit Fenster und einem "Seitenfeld" für die Angabe der Adresse und Bankverbindungen. Die Schriftgrößen für den Ausdruck können für jeden dieser vier Bereiche getrennt im Druckertreiber festgelegt werden (s.u.).

Eine sehr hilfreiche Funktion von WRITER ST ist die Makrodefinition. Der Benutzer kann sich unter einem sechsstelligen Kürzel verschiedene Textmakros definieren, die mit Aufruf des entsprechenden Kürzels automatisch in den Text eingefügt werden.

Die Makros werden in einer eigenen Makrodatei gespeichert, die bis zu 32000 Makros enthalten kann. Ein Makro kann maximal 160 Textzeichen enthalten. Falls das nicht reichen sollte, kann man Makros auch verketten, d.h. aus einem Makro heraus den Text eines anderen Makros aufzurufen. Ein Makro kann außerdem Steuerzeichen enthalten für Zeilenumbruch, Sprung zum nächsten Tabulatorstop sowie Fettdruck, Unterstreichen oder Hoch- und Tiefstellen von Text.

Für den Fall, daß man einmal den Namen eines Makros vergessen haben sollte, gibt es eine sehr gute Hilfemöglichkeit (direkt über die HELP-Taste). Ein Fenster mit einem Ausschnitt der Makrodatei (28 Kürzel und die ersten 40 Textstellen der dazugehörigen Makros) wird angezeigt (Abb.5), bei größeren Makrodateien kann sequentiell geblättert oder gezielt durch Angabe der ersten beiden Buchstaben des Kürzels gesucht werden. In diesem Fenster kann man auch einzelne Makros löschen.

Innerhalb eines Dokuments können bis zu vier verschiedene Tastaturbelegungen gleichzeitig angesprochen werden. Mitgeliefert sind Standard (Deutsch), Englisch, Französisch, US-Englisch, Mathematik und Chemie. Jede Tastaturbelegung besteht aus drei Ebenen, deren Zeichen normal, mit der SHIFT- oder der CapsLock-Taste angesprochen werden. Dazu kann man sich mit einem Hilfsprogramm eigene Tastaturbelegungen zusammenstellen. Das funktioniert mit einem Hilfsprogramm, das außerhalb von WRITER ST aufgerufen wird. Auch eine eigene Tastaturbelegung hat drei Ebenen. Jede Taste (außer den Steuertasten ESC, TAB, CTRL, SHIFT, ALT, Backspace, RETURN und ENTER) kann jedem darstellbaren Zeichen zugeordnet werden.

Das Prinzip der Druckertreiber entspricht ungefähr dem von 1st Word. Druckertreiberdateien können als ASCII-Dateien selbst erstellt und mit einem Hilfsprogramm installiert werden. Einige Druckertreiber sind bereits im Lieferumfang enthalten. Getestet wurde der Treiber für den NEC P6/7. Leider werden die schönen Fonts, die es für die Bildschirmdarstellung gibt (gotik, script, blippo...), beim

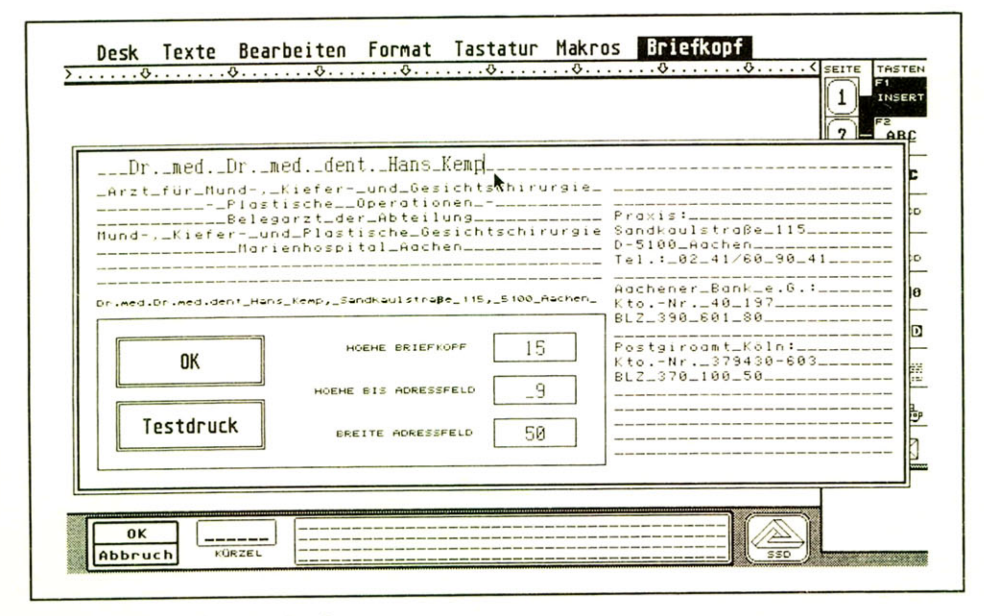


Bild 4: Definition eines Briefkopfes

geht weiter...

Ausdruck nicht berücksichtigt. In dem Auswahlmenü für den Druck werden fünf verschiedene Schrifttypen angeboten: Pica und Elite im Draft-Modus (LQ10 und LQ12), kursiv und Pica in Schönschrift und Doppelte Größe. Benutzt man den mitgelieferten Druckertreiber für den NEC, werden Texte auch bei Draft-Einstellung in Letter-Quality (nicht in Proportionalschrift) ausgedruckt. Eckige Klammern werden beim Drucken zu 'Ä' und 'Ü', andere Sonderzeichen werden gar nicht ausgedruckt. Die Voreinstellung für den Ausdruck des Briefkopfes ist auch nicht ganz gelungen, die Schrift für die Kopfzeile ist viel zu breit, ein etwas längerer Name paßt nicht mehr in eine Zeile. Zum Glück kann man den Druckertreiber ja selbst ändern!

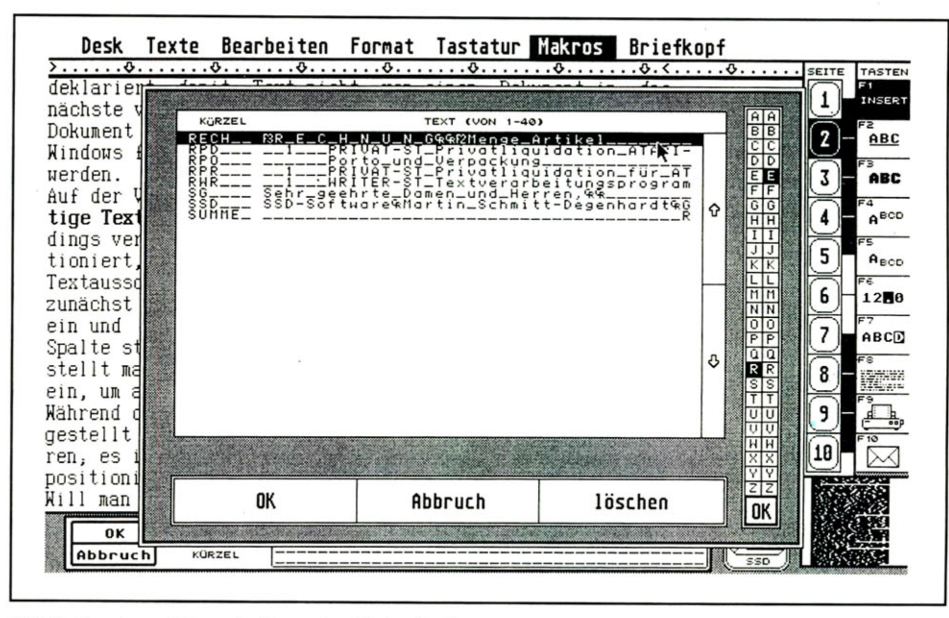


Bild 5: Fenster mit Ausschnitt aus der Makrodatei

Zusammenfassung

Gut sind bei WRITER ST einige Zusatzfunktionen wie Makroverwaltung, Definition von Formularen und einem eigenen Briefkopf. Praktisch sind auch das Speichern der eigenen Arbeitsumgebung mit automatischem Einladen eines oder mehrerer Texte und das Arbeiten mit mehreren Tastaturbelegungen. Praktisch, um sich einen Überblick über das Seitenlayout zu verschaffen, ist auch das kleine Fenster in der unteren Ecke den Bildschirms, das eine stark verkleinerte Abbildung der kompletten aktuellen Seite darstellt. Gut sind auch die Möglichkeit, Komandos über die Tastatur einzugeben und das schnelle Blättern zwischen den Seiten, wodurch die Arbeitsgeschwindigkeit erhöht werden kann.
Mängel weist das Programm aber bei herkömmlichen Funktionen der Textverarbeitung auf. Ein neues Dokument kann nur über Umwege erstellt werden, ein Dokument kann nicht länger als 10 Seiten sein, Kopf- und Fußzeilen, sowie Seitennumerierung fehlen vollständig. Ein Dokument kann maximal 70 Zeichen breit sein und der Zeilenabstand (einzeilig) kann nicht verändert werden. Zur Verringerung der Arbeitsgeschwindigkeit trägt vor allem das umständliche Verhalten von WRITER ST bei, wenn man eingegebenen Text nachträglich edieren oder etwas einfügen möchte. Es fehlen Blockmarkierungen, die nicht an rechteckige Bereiche gebunden sind, so daß es möglich wäre, einen Satz zusammenhängend zu bearbeiten (kopieren, verschieben, löschen). Auch bei dem NEC-Druckertreiber ist deutlich zu sehen, daß es sich bei dem getesteten Programm noch um Version 1.1 handelt. Schade, daß es offensichtlich noch nicht möglich ist, ein Textverarbeitungsprogramm mit guten Edierfunktionen und mit einigen in WRITER ST enthaltenen Zusatzfunktionen anzubieten.

Hardware-Vorraussetzungen, Lieferumfang und Preis

Hardware:
Alle Atari STs mit mindestens 512 KB (mit ROM-TOS), doppelseitigem Laufwerk und Monochrommonitor.

Lieferumfang:
Eine doppelseitig beschriebene Diskette mit dem eigentlichen Textverarbeitungsprogramm, Druckertreibern für Riteman F+, Epson FX-kompatible, Brother M-4018, Epson LQ-850, Epson LQ-1050, Epson FX, NEC P6 und Seikosha SP-800, ein Druckerinstallationsprogramm, ein Reorganisationsprogramm für die Makrodatei und ein Tastaturbelegungsprogramm. 64 Seiten lange deutschsprachige Dokumentation DIN A5

Cordula Vogler

Preis: DM 98,-

Bezugsadresse:
SSD-Software
Gregorstr. 1
5100 Aachen

ENDE



Programmier Praxis

Auch diesmal bietet die Programmierpraxis wieder ein breites Spektrum, in dem für jeden etwas dabei sein sollte. (heute vor allem für unsere PASCAL-Freunde). Dies ist vor allem den fleißigen Programmierern zu verdanken, die ihre Programme und Programmroutinen hier in der Programmierpraxis dem Leserkreis der ST-Computer zur Verfügung stellen. Ein herzliches Dankeschön an all diese und die, die sich in Zukunft an dieser Ideenbörse beteiligen werden.

Doch nun zu den verschiedenen Programmen. Zuerst haben wir da eine VAL-Routine. Wer sich irgendwann mal mit einem BASIC-Dialekt beschäftigt hat (und wer hat das nicht?), kennt mit Sicherheit den VAL-Befehl, der einen String in eine numerische Variable umwandelt.

Sehr schön ist auch eine kleine Routine, mit der sich Menüleisten nicht nur am oberen Bildschirmrand abbilden lassen. Turtle-Grafik in PASCAL ist sicherlich auch eine brauchbare Anwendung, die viele Interessenten finden wird.

Doch auch für die C- und BASIC-Freunde ist etwas dabei. Wen die etwas ungeschickte Position der Größer-/Kleiner-Zeichen schon immer gestört hat, dem wird die C-Routine zur Umbelegung der Taste sicher gefallen.

In Omikron-BASIC ist eine Routine zum Zeichnen von Achsenkreuzen gesch-rieben, die in Funktionsplot-Programmen bestimmt eine gute Anwendung findet.

INHALT

VAL in PASCAL

- Umwandlung von Strings in numerische Variablen

Als ich vor einiger Zeit anfing Mathematikprogramme zu schreiben, mußte ich feststellen, daß mir CCD Pascal+ nicht alle nötigen Hilfsmittel zur Verfügung stellte. Ich setzte mich also hin und schrieb zunächst eine VAL-Routine, wie sie mir von Basic aus bekannt war. Diese Routine ist als Include-Datei ausgelegt.

Die dazuge-hörige Datei hat den Namen VALUE.INC. Sie macht aus einem übergebenen Stringwert einen Realwert, falls dies möglich ist. Sollte eine Umwandlung nicht möglich sein, so enthält die global zu deklarierende Variable 'Fehler', die vom Typ Boolean sein muß, den Wert 'FALSE', ansonsten ist sie 'TRUE'. Ich bin bei dieser Routine einige Kompromisse in puncto Codelänge und Rechengeschwindigkeit eingegangen, wodurch sich eine Zeit von ca. 10 ms für die Umwandlung eines 13-stelligen Strings(-123.4567E-16) ergibt.

Das Prinzip der Umwandlung beruht auf der systematischen Zerpflückung des Strings und ist an und für sich leicht zu durchblicken. Das Programm CHR_REAL.PAS ist ein kleines Demoprogramm für VALUE.INC und wandelt die eingegebenen Strings in Realwerte um, bis man als String den Wert 'ENDE' eingibt. In das Programm CHR_REAL ist noch eine Procedure eingebaut, die alle Kleinbuchstaben in Großbuchstaben umwandelt, wodurch es egal wird, wie die Tastatur gerade eingestellt ist. Die Value-Routine wird folgendermaßen angewendet:

VALUE.INC

```
{Value-Routine von Klaus Wilczek für ST Pascal+}
{wanden von Strings in Realwerte}

FUNCTION VAL(X:STRING):REAL;

VAR I1,I2:STRING;
    TESTWORD:STRING;
    N,LE,E1,R,EX,KO,VM,VE:INTEGER;
    M,I3,I4:REAL;

PROCEDURE INT(R:INTEGER);{Erstellen der Zahlenwerte}
BEGIN
  IF FEHLER<>FALSE THEN
  BEGIN
    I3:=0;
    IF KO=0 THEN
      FOR R:= 1 TO LENGTH(I1) DO
        I3:=I3*10 + ORD(I1[R])-48
    ELSE
    BEGIN
      FOR R:= 1 TO (KO-1) DO
        I3:=I3*10 +  ORD(I1[R])-48;
      I4:=0;
      FOR R:= LENGTH(I1) DOWNTO (KO+1) DO
        I4:=(I4+ORD(I1[R])-48)/10;
      I3:=I3+I4;
    END;
  END;
END;

PROCEDURE SYNTAXCHECK(TESTWORD:STRING;KO:INTEGER);
{Syntaxkontrolle}
VAR I,W:INTEGER;
BEGIN
  IF KO >0 THEN DELETE (TESTWORD,KO,1);
  IF TESTWORD='' THEN FEHLER:=FALSE ELSE
  FOR I:=1 TO LENGTH(TESTWORD)DO
  BEGIN
    W:=ORD(TESTWORD[I])-48;
    IF (W<0) OR (W>9) THEN FEHLER:=FALSE;
  END;
END;

BEGIN
  FEHLER:=TRUE;
  REPEAT{löschen der Leerstellen}
    N:=POS(' ',X);
    IF (N=1) THEN DELETE(X,N,1);
  UNTIL (N>1) OR (N=0);
  IF (POS(' ',X)>0) THEN DELETE(X,N,LENGTH(X)-N+1);
  IF POS('-',X)=1 THEN{suchen der Vorzeichen}
    BEGIN
      VM:=-1;
      DELETE(X,1,1)
    END
  ELSE VM:=1;
  KO := POS('.',X);
  EX := POS('E',X);
  IF EX = 0 THEN
  BEGIN {herstellen von Mantisse und Exponent}
    I1:=COPY(X,1,LENGTH(X));
    I2:='0';
  END
  ELSE
  BEGIN
    I1:=COPY(X,1,EX-1);
    I2:=COPY(X,EX+1,LENGTH(X)-EX);
    VE:=1;
    IF (I2[1]='-') THEN VE:=-1;
    IF (VE=-1) OR (I2[1]='+') THEN DELETE(I2,1,1);
  END;
  TESTWORD:=CONCAT(I1,I2);
  SYNTAXCHECK(TESTWORD,KO);
  INT(R); M:=I3; I1:=I2;
  KO:=0; INT(R);
  E1:=TRUNC(I3);
  IF VE = 1 THEN VAL:=(M*VM)*PwrOfTen(E1)
  {herstellen der Realwerte}
  ELSE VAL:=(M*VM)*(1/PwrOfTen(E1));
  IF FEHLER=FALSE THEN VAL:=0;
END;
```

```
dest := val(source);
dest   - Real
source - String
```

Die Routine führt vor der Umwandlung eine grobe Syntaxkontrolle durch, wodurch die Gefahr eines eventuell auftretenden Fehlers sehr gering sein dürfte. Wird die Datei in irgendwelche Programme eingefügt, lassen sich z.B. Eingaben schnell auf ihre Richtigkeit prüfen, wodurch sich viele Fehler vermeiden lassen.

Weiterhin war ich seit einiger Zeit auf der Suche nach einem guten Algorithmus für einen (*)Funktionsinterpreter. Als ich dann auf die Zeitschrift MC, Ausgabe 5/87 aufmerksam gemacht wurde, beschloß ich, den dort in Turbo-Pascal geschriebenen FI(*) in ST Pascal+ umzuschreiben. Hierbei zeigten sich die Differenzen zwischen den beiden Programmiersprachen doch recht deutlich, es mußte jedoch nicht allzuviel umgeändert werden. Das Resultat war ein recht gut und schnell funktionierender FI. Hier wird ebenfalls die VAL-Routine benutzt, die ja ansonsten bei Turbo-Pascal bereits zur Verfügung steht.

Die Aufgabe des FI ist es, einen beliebigen Funktionstherm als String einzulesen, diesen zu analysieren, auf Syntaxfehler zu kontrollieren und zuzulassen, daß beliebige X-Werte eingesetzt werden können bzw. die dazugehörigen Funktionswerte errechnet werden.

Ein FI muß die üblichen Rechenregeln beherrschen:

- Punkt- vor Strichrechnung
- Klammern zuerst
- von links nach rechts

Hieraus ergibt sich eine Hierarchie der Rechenoperatoren:

1. Klammern und Funktionen
2. Potenz ^
3. Vorzeichen +,-
4. Punktrechnung *, /
5. Strichrechnung +,-

Aus der Funktion wird hier ein Baum gebaut, wobei die Knoten des Baumes durch Operatoren, Funktionen, Vorzeichen, Zahlen und Variablen belegt sind. Die Knoten sind als variante Records deklariert. Der Baum wird von den höherwertigen Operationen hin zu den niederwertigeren durchgearbeitet. Das Programm FUNKTION.TOS erlaubt es, Funktionen einzugeben und Funktionswerte zu berechnen. Leider mußte ich feststellen, daß die mathematischen Funktionen von ST Pascal+ sehr ungenau sind(z.B. 2^16=65535.99..), und ich will dies noch ändern. Alle bisher angebotenen Algorithmen sind aber entweder selber zu ungenau oder zu umfangreich programmiert, so daß ich es selber probieren will.

Mit diesen beiden Funktionen kann bestimmt jeder, der irgendwann einmal mathematische Programme schreiben will, etwas anfangen. Die Programme sollten als TOS-Anwendungen compiliert werden, da ich auf Bildschirmaufbau keinen großen Wert gelegt habe.

Klaus Wilczek

CHR_REAL.PAS

```
{Programm zum testen von VALUE.INC geschrieben von
 Klaus Wilczek}

PROGRAM TEST_VALUE;

VAR X:STRING;
    FEHLER:BOOLEAN; {globale Deklaration der Variablen
                     Fehler}

{$I VALUE.INC}

PROCEDURE UPPER;{Umwandlung aller Kleinbuchstaben in
                 Gro_buchstaben}
{Die Variable (hier X) mu_ jeweils angepasst werden}

VAR I:INTEGER;

BEGIN
  FOR I:=1 TO LENGTH(X) DO
   IF (ORD(X[I])>96) AND (ORD(X[I])<123) THEN
   BEGIN
     INSERT(CHR(ORD(X[I])-32),X,I);
     DELETE(X,I+1,1);
   END;
END;


BEGIN
  WRITELN('Beenden des Programms mit "Ende" !');
  REPEAT
    WRITELN;WRITE('Bitte einen Wert eingeben : ');
    READLN(X);
    UPPER;{ alle Kl.bst. werden umgewandelt in Gr.bst.}
    IF X<>'ENDE' THEN
    BEGIN
      WRITELN ; WRITELN('  X als Stringwert : ',X);
      WRITELN ; WRITELN('- Umgewandelt zu einem
                         Realwert: ',VAL(X));
      WRITELN ; WRITELN('  Fehlervariable : ',FEHLER);
    END;
  UNTIL X='ENDE';{aussteigen wenn X='ENDE'}
END.
```

FUNKTION.PAS

```
PROGRAM FORMELINTERPRETER;

TYPE ARTTYP       = (VORZEICHEN,OPERATOR,FUNKTION,ZAHL,
                     VARX);
     STRING128    = STRING[128];
     OPTYP        = SET OF CHAR;
     PTR          = ^KNOTENTYP;
     KNOTENTYP    = RECORD CASE ART:ARTTYP OF
                      VORZEICHEN : (VON:PTR);
                      OPERATOR   : (OP:CHAR;
                                    LINKS,RECHTS:PTR);
                      FUNKTION   : (FN:STRING[6];
                                    NEXT:PTR);
                      ZAHL       : (ZA:REAL);
                      VARX       : (X:REAL);
                    END;

VAR X,Y    : REAL;
    WAHL   : CHAR;
    FKT    : PTR;
    TERM   : STRING128;
    FEHLER : BOOLEAN;

{$I VALUE.INC}

FUNCTION SUCHEOPERATOR (OPS:OPTYP; VAR K:INTEGER;
         TERM:STRING128):BOOLEAN;
         VAR KLAMMER:INTEGER;GEFUNDEN:BOOLEAN;
BEGIN
  GEFUNDEN:=FALSE;KLAMMER:=0;K:=LENGTH(TERM)+1;
  REPEAT
    K:=K-1;
    IF TERM[K]='('THEN KLAMMER:=PRED(KLAMMER);
    IF TERM[K]=')'THEN KLAMMER:=SUCC(KLAMMER);
    IF (KLAMMER=0)AND(TERM[K]IN OPS)AND(K>1)
     AND(NOT(TERM[K-1]IN['E','^']))THEN GEFUNDEN:=TRUE;
  UNTIL GEFUNDEN OR (K=1);
  SUCHEOPERATOR:=GEFUNDEN;
END;

FUNCTION SUCHEPLUSMINUS(VAR K:INTEGER;TERM:STRING128):
                                              BOOLEAN;
```

```
BEGIN
    SUCHEPLUSMINUS:=SUCHEOPERATOR(['+','-'],K,TERM);
END;

 FUNCTION SUCHEMALDURCH(VAR K:INTEGER;TERM:STRING128):
                                             BOOLEAN;
BEGIN
    SUCHEMALDURCH:=SUCHEOPERATOR(['*','/'],K,TERM);
END;

 FUNCTION SUCHEVORZEICHEN(TERM:STRING128):BOOLEAN;
BEGIN
   SUCHEVORZEICHEN:=(TERM[1] IN ['+','-']);
END;

 FUNCTION SUCHEPOTENZ(VAR K:INTEGER;TERM:STRING128):
                                             BOOLEAN;
BEGIN
    SUCHEPOTENZ:=SUCHEOPERATOR(['^'],K,TERM);
END;

 FUNCTION SUCHEFUNKTION(VAR K:INTEGER;TERM:STRING128):
                                             BOOLEAN;
  VAR F:STRING128;
BEGIN
   SUCHEFUNKTION:=FALSE;K:=POS('(',TERM);
  IF K>0 THEN
  BEGIN
     F:=COPY(TERM,1,K-1);
       IF((F='ABS')OR(F='ARCTAN')OR(F='COS')OR(F='EXP')OR
                                             (F='FRAC')OR
       (F='INT')OR(F='LN')OR(F='SIN')OR(F='SQR')OR
                                             (F='SQRT')OR
       (F='TAN')OR(F=''))AND(TERM[LENGTH(TERM)]=')')
     THEN SUCHEFUNKTION:=TRUE
  END;
END;

 FUNCTION SUCHEZAHL(VAR WERT:REAL; TERM:STRING128):
                                             BOOLEAN;
BEGIN
  WERT:=VAL(TERM);
  SUCHEZAHL:=FEHLER;
END;

 FUNCTION SUCHEX(TERM:STRING128):BOOLEAN;
BEGIN
  SUCHEX:=(TERM='X');
END;

 FUNCTION FUNKTIONSANALYSE(TERM:STRING128):PTR;

  VAR TERMOK:BOOLEAN;FKT:PTR;

  PROCEDURE BAUEBAUM( VAR KNOTEN:PTR; TERM:STRING128);
   VAR WERT:REAL; K:INTEGER; VZ:CHAR;
  BEGIN
     IF TERMOK AND (LENGTH(TERM)>0) THEN
       IF SUCHEPLUSMINUS(K,TERM) THEN BEGIN
         NEW(KNOTEN);
           KNOTEN^.ART:=OPERATOR; KNOTEN^.OP:=TERM[K];
           BAUEBAUM(KNOTEN^.LINKS ,COPY(TERM,1,K-1));
         BAUEBAUM(KNOTEN^.RECHTS,
                    COPY(TERM,K+1,LENGTH(TERM)-K));
       END
       ELSE IF SUCHEMALDURCH(K,TERM) THEN BEGIN
         NEW(KNOTEN);
           KNOTEN^.ART:=OPERATOR; KNOTEN^.OP:=TERM[K];
           BAUEBAUM(KNOTEN^.LINKS ,COPY(TERM,1,K-1));
         BAUEBAUM(KNOTEN^.RECHTS,
                    COPY(TERM,K+1,LENGTH(TERM)-K));
       END
       ELSE IF SUCHEVORZEICHEN(TERM) THEN BEGIN
         VZ:=TERM[1]; DELETE(TERM,1,1);
         CASE VZ OF
           '+': BAUEBAUM(KNOTEN, TERM);
           '-': BEGIN
                  NEW(KNOTEN); KNOTEN^.ART:=VORZEICHEN;
                  BAUEBAUM(KNOTEN^.VON, TERM);
                END;
         END
       END
       ELSE IF SUCHEPOTENZ(K,TERM) THEN BEGIN
         NEW(KNOTEN);
           KNOTEN^.ART:=OPERATOR; KNOTEN^.OP:=TERM[K];
           BAUEBAUM(KNOTEN^.LINKS ,COPY(TERM,1,K-1));
         BAUEBAUM(KNOTEN^.RECHTS,
                    COPY(TERM,K+1,LENGTH(TERM)-K));
       END
       ELSE IF SUCHEFUNKTION(K,TERM) THEN BEGIN
         NEW(KNOTEN);
          KNOTEN^.ART:=FUNKTION;
          KNOTEN^.FN:=COPY(TERM,1,K-1);
         BAUEBAUM(KNOTEN^.NEXT,
                    COPY(TERM,K+1,LENGTH(TERM)-1-K));
       END
       ELSE IF SUCHEZAHL(WERT,TERM) THEN BEGIN
         NEW(KNOTEN);
          KNOTEN^.ART:=ZAHL; KNOTEN^.ZA:=WERT;
       END
      ELSE IF SUCHEX(TERM) THEN BEGIN
         NEW(KNOTEN);
          KNOTEN^.ART:=VARX;
       END
       ELSE TERMOK:=FALSE
     ELSE TERMOK:=FALSE;
  END;

  PROCEDURE UPPER;
  VAR I:INTEGER;
  BEGIN
     FOR I:=1 TO LENGTH(TERM) DO
       IF (ORD(TERM[I])>96) AND
          (ORD(TERM[I])<123) THEN
       BEGIN
          INSERT(CHR(ORD(TERM[I])-32),TERM,I);
         DELETE(TERM,I+1,1);
       END;
  END;

BEGIN
  TERMOK:=TRUE; UPPER;
  BAUEBAUM(FKT,TERM);
  IF NOT TERMOK
    THEN FUNKTIONSANALYSE:=NIL
    ELSE FUNKTIONSANALYSE:=FKT;
END;

 FUNCTION FUNKTIONSBERECHNUNG(FKT:PTR;X:REAL):REAL;

CONST MAXREAL=1.0E30; MINREAL=1.0E-30;
VAR   WERTOK:BOOLEAN;

 FUNCTION WERT(ZEIGER:PTR):REAL;
  VAR TEST,FX:REAL;
BEGIN
  FX:=X;
  IF WERTOK THEN
    WITH ZEIGER^ DO
      CASE ART OF
        VORZEICHEN: WERT:=-WERT(VON);
          OPERATOR:CASE OP OF
                   '+':WERT:=WERT(LINKS)+WERT(RECHTS);
                   '-':WERT:=WERT(LINKS)-WERT(RECHTS);
                   '*':WERT:=WERT(LINKS)*WERT(RECHTS);
                   '/':BEGIN
                         TEST:=WERT(RECHTS);
                         IF ABS(TEST)>MINREAL
                           THEN WERT:=WERT(LINKS)/TEST
                           ELSE WERTOK:=FALSE;
                       END;
                   '^':BEGIN
                         TEST:=WERT(LINKS);
                         IF TEST>MINREAL
                           THEN WERT:=EXP(WERT(RECHTS)*
                                      LN(TEST))
                           ELSE WERTOK:=FALSE;
                       END;
                 END;
          FUNKTION: BEGIN
                      IF FN='' THEN WERT:=WERT(NEXT);
                      IF FN='ABS' THEN
                       WERT:=ABS(WERT(NEXT));
                      IF FN='ARCTAN'THEN
```

```
                         WERT:=ARCTAN(WERT(NEXT));
                        IF FN='COS' THEN
                         WERT:=COS(WERT(NEXT));
                        IF FN='EXP' THEN
                         WERT:=EXP(WERT(NEXT));
                        IF FN='LN' THEN BEGIN
                            TEST:=WERT(NEXT);
                              IF TEST>0
                              THEN WERT:=LN(TEST)
                              ELSE WERT OK:=FALSE;
                                          END;
                        IF FN='SIN' THEN
                         WERT:=SIN(WERT(NEXT));
                        IF FN='SQR' THEN
                         WERT:=SQR(WERT(NEXT));
                        IF FN='SQRT' THEN BEGIN
                              TEST:=WERT(NEXT);
                                 IF TEST>0
                                 THEN WERT:=SQRT(TEST)
                                 ELSE WERT OK:=FALSE;
                                        END;
                         IF FN='TAN'  THEN WERT:=SIN(WERT
                         (NEXT))/COS(WERT(NEXT));
                       END;
                  ZAHL: WERT:=ZA;
                  VARX: WERT:=FX;
                  END;
           END;

BEGIN
  IF FKT<>NIL THEN BEGIN
     WERTOK:=TRUE; FUNKTIONSBERECHNUNG:=WERT(FKT);
    IF NOT WERTOK THEN FUNKTIONSBERECHNUNG:=MAXREAL
   END
   ELSE FUNKTIONSBERECHNUNG:=MAXREAL;
END;

PROCEDURE clear_line; BEGIN write (chr (27), 'l') END;
PROCEDURE gotoxy (x, y: integer);
BEGIN IF x < 0 THEN x := 0
                ELSE IF x > 79 THEN x := 79;
         IF y < 0 THEN y := 0
                ELSE IF y > 24 THEN y := 24;
         write (chr (27), 'Y', chr (y + 32), chr (x +
32))
END;
BEGIN
  REPEAT
      GOTOXY(3,2);WRITE('Interative Funktionseingabe');
     GOTOXY(3,4);WRITE('<1> Term eingeben');
     GOTOXY(3,5);WRITE('<2> Funktionswerte');
     GOTOXY(3,6);WRITE('<3> Ende');
    REPEAT
       GOTOXY(7,9);CLEAR_LINE;WRITE('Ihre Wahl: ');
      READ(WAHL);CLEAR_LINE;
     UNTIL WAHL IN ['1'..'3'];GOTOXY(7,9);
    CASE WAHL OF
      '1': BEGIN
             WRITE('Funktionsterm: f(x)= ');
             READLN(TERM);CLEAR_LINE;
             FKT:=FUNKTIONSANALYSE(TERM);
            IF FKT=NIL THEN BEGIN
               GOTOXY(7,11);WRITE('FEHLER IN TERM');
              READLN;CLEAR_LINE;END;
           END;
       '2': BEGIN
             IF FKT<>NIL THEN BEGIN
               WRITE('X = ');
                GOTOXY(11,9);READLN(X);CLEAR_LINE;
                Y:=FUNKTIONSBERECHNUNG(FKT,X);
                GOTOXY(0,11);CLEAR_LINE;GOTOXY(11,11);
               WRITELN('f(X)= ',Y);
              END
            ELSE BEGIN
                    GOTOXY(7,11);CLEAR_LINE;
                    WRITE('KEINE FUNKTION DA.');
                 END;
            READLN;
          END;
      END;
  UNTIL WAHL='3';
END.
```

FLIP

Menüleisten einmal anders

Wie oft begegnet einem diese Situation: Man klickt auf dem Bildschirm etwas an und muß in die Menüleiste, um die Option zu spezifizieren, damit sie ausgelöst werden kann.

Beim KUMA-Resource-Editor ist das ganz anders. Sobald man hier etwas anklickt, erscheint bei der Maus ein kleines Flipmenü, um einem den Weg in die Menüleiste zu ersparen. Die hier vorgestellte Routine leistet eben dieses: Sie kann ein Flipmenü an jedem Bildschirmpunkt (in der Cursor-Matrix 80x25) plazieren.
Diese Menüs werden nicht mit umständlichen GEM-Einträgen gestartet, sondern sind universell einsetzbar, sie initialisieren sich bei jedem Aufruf von selbst. Zur Einführung einige Worte zur...

Arbeitsweise

Die Funktion bekommt beim Aufruf die Texte der Menüpunkte sowie die gewünschte Bildschirmposition des Menüs übergeben. Nach der Ausgabe auf dem Bildschirm wird nun ständig die Position der Maus und der Status der Maustasten abgefragt. Jenachdem wo sich die Maus befindet, wird der unter ihr liegende Menüpunkt invertiert. Befindet sich die Maus außerhalb des Menüs, wird nichts invertiert.
Werden die Maustasten aktiviert (rechts, links oder beide, das spielt keine Rolle), wird der invertierte Menüpunkt angewählt und an die aufrufende Prozedur die entspr. Eintragsnummer weitergegeben. Außerdem wird gemeldet, ob der Eintrag mit der rechten, linken oder gar mit beiden Maustasten angewählt wurde.

Die Umgebung

Von einer Funktion dieser Art verlangt man, daß nach deren Aufruf der Bildschirm wieder im urspr. Zustand vorliegt. Dies bedeutete für mich, der Einfachheit halber den gesamten Bildschirm zwischenzuspeichern. Entsprechendes leisten die Prozeduren Save_Screen und Restore_Screen (Z. 64-77), die wiederum bestimmte Typen und Variablen sowie die XBIOS-Funktion LogBase (XBIOS(2)) benötigen.
Um die Mausposition dauernd feststellen zu können, bediene ich mich der AES-Funktion 79: Graf_MKState. Auch sie benötigt spezielle Variablen und Typen (im Listing als solche gekennzeichnet).
Die Prozeduren write_xy und hiwrite_xy (Z. 78-86) dienen ebenso zum Darstellen des Menüs auf dem Bildschirm wie die Prozedur Rahmen (Z.88-100). All diese Prozeduren setzen ihre benötigten x-/y-Koordinaten als Koordinaten aus der Snap-Matrix (also 80x25) voraus.
Alle o.g. Prozeduren sind schon als solche brauchbar und nützlich in jedem Programm. Die eigentliche Funktion "FLIP" (Z.105-188) setzt sie als vereinbart voraus. Aus diesem Grund sind sie auch mit aufgelistet.

Die Syntax der Funktion

Die Funktionsvereinbarung lautet:
FUNCTION Flip(x,y : INTEGER; was : Str255; VAR M_Keys : INTEGER)
: INTEGER;
Die Variablen x und y legen die obere linke Ecke des Menüs in der 80x25-Bildschirmmatrix fest. Im String "Was" finden sich die Menüeinträge, die jeweils durch einen Underscore getrennt sind.
Ein Beispiel: Man will ein Menü mit folgendem Aussehen erzeugen:

```
Mami
Papi
 Ich
 und
 Du
```

Dann muß der String so lauten:
`"Mami_Papi_Ich_und_Du".`
Weitere Beispiele finden sich im Hauptprogramm des Listings. Jeder Menüpunkt kann eine Länge von max. 40 Zeichen haben, der gesamte Definitionsstring darf max. 255 Zeichen beinhalten. Die letztgenannte Grenze ist durch das Pascal gesetzt.
In der Variablen M_Keys schließlich wird der Status der Maustasten zurückgegeben. Hierbei bedeutet 1, daß die linke Maustaste gedrückt wurde; 2 ist es bei der rechten, und bei beiden Tasten gleichzeitig ist es 3.
Zur Funktion selbst ist eigentlich nichts zu erwähnen. Die verschiedenen Unterprozeduren sind erstens sehr einfach und zweitens im Listing näher erläutert. So

Geht weiter...

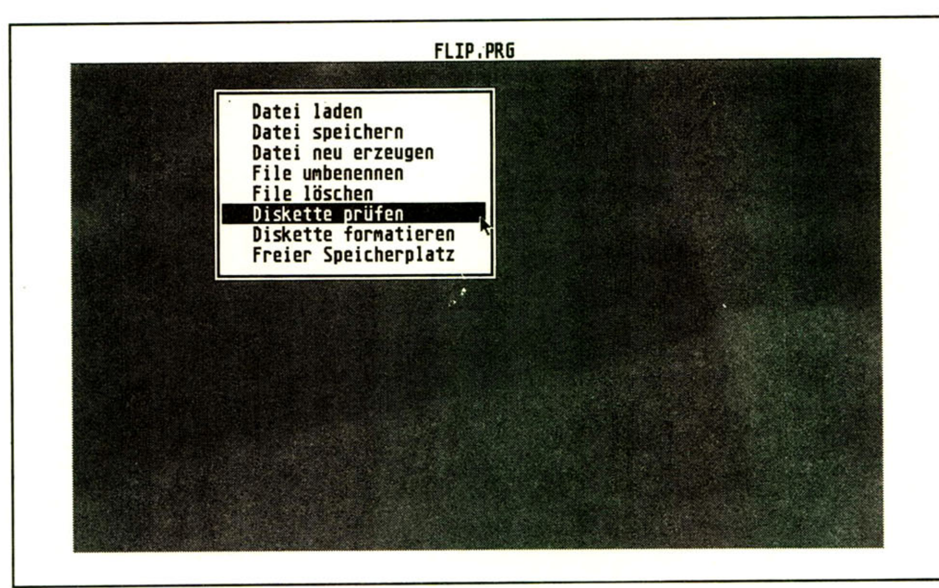


bleibt zum Schluß nur noch ein...

Ausblick

Die hier vorgestellte Flipmenü-Routine ist keineswegs vollständig. Sie stellt vielmehr eine Minimalversion dar. Denkbar wären z.B. DISABLE-te Menüeinträge (die z.B. durch ein anderes Trennzeichen als dem Underscore, z.B. den Klammeraffen) kenntlich gemacht werden könnten. In diesem Fall müßten die Prozeduren write_xy und hiwrite_xy durch entspr. Draw_String-Anweisungen ersetzt werden (was jedoch mit einem Geschwindigkeitsverlust einhergeht).
Ein weiterer Vorschlag wäre, das Menü gleichzeitig auch über die Tastatur steuern zu können, um z.B. bestimmte Einträge mit den Funktionstasten anwählen zu können. All diese Möglichkeiten sind auf der Basis des gelieferten "Rohlistings" relativ einfach "nachzurüsten". Solche Versionen der Funktion sind bei mir mittlerweile erfolgreich im Einsatz. Sie haben allerdings den Nachteil, den Rahmen dieser Programmierecke zu sprengen, weshalb ich mich auf die einfachste Version, die aber in den meisten Fällen durchaus ausreichend sein wird, beschränkt habe.
Das hier vorliegende Programm wurde mit ST-Pascal Plus 1.20 erstellt. Wer das Glück hatte, bereits mit einem Update versorgt zu werden, wird mit Sicherheit die eine oder andere Routine deutlich verkürzen oder sogar wegfallen lassen können.

Udo Hilwerling



Beispiel für eine verschobene Menüleiste



```
{$D-, S10} { Fehlermeldungen ausschalten }
PROGRAM Flipmenue;

  { Mausgesteuertes Flipmenue an beliebiger
    Bildschirmposition.
    Die aufgefuehrten Variablen und Prozeduren gehoeren
    zur benoetigten
    Umgebung der Routine.
    Autor: Udo Hilwerling,
           Zumlohstr. 21
           4410 Warendorf 1  }

CONST
  {$I GEMCONST.PAS }

TYPE
  {$I GEMTYPE.PAS              }

  { Typen fuer AES-Aufruf }
  AES_Pointer     = ^CHAR;
  Int_In_Parms    = ARRAY[0..15] OF INTEGER;
  Int_Out_Parms   = ARRAY[0..45] OF INTEGER;
  Addr_In_Parms   = ARRAY[0..1]  OF AES_Pointer;
  Addr_Out_Parms  = ARRAY[0..0]  OF AES_Pointer;

  { Typen zum Zwischenspeichern des Bildschirms }
  Screen          = PACKED ARRAY[1..32000] OF BYTE;
  ScreenPointer = ^Screen;

VAR
  { Variablen fuer AES-Aufruf }
  int_in         : Int_In_Parms;
  int_out        : Int_Out_Parms;
  addr_in        : Addr_In_Parms;
  addr_out       : Addr_Out_Parms;

  ScreenBase   : ScreenPointer;  { Zeiger auf Bildschirm
                                   speicher       }
  Schirm       : Screen;         { Hier wird der
                                   Bildschirm gepuffert }

  test, mouse : INTEGER;         { Testvariablen }
  hilf        : Str255;

{$I GEMSUBS.PAS               }

{ AES-Prozeduren
  ----------------------------}
PROCEDURE AES_Call(op : INTEGER;
           VAR int_in   : Int_In_Parms;
           VAR int_out  : Int_Out_Parms;
           VAR addr_in  : Addr_In_Parms;
           VAR addr_out : Addr_Out_Parms); EXTERNAL;

PROCEDURE Graf_MKState(VAR x,y, pr, key : INTEGER);
  BEGIN
    AES_Call(79, int_in, int_out, addr_in, addr_out);
    x:=int_out[1]; y:=int_out[2];
    pr:=int_out[3];
    key:=int_out[4]
  END;

{----------------------------}

FUNCTION LogBase : ScreenPointer;
  XBIOS(2);

PROCEDURE Save_Screen; { Bildschirm zwischenspeichern }
  BEGIN
    Hide_Mouse;
    {$P-} Schirm:=ScreenBase^; {$P=}
    Show_Mouse;
```

```
    END;

  PROCEDURE Restore_Screen; { Bildschirm
                              wiederherstellen }
    BEGIN
      Hide_Mouse;
      {$P-} ScreenBase^:=Schirm; {$P=}
      Show_Mouse;
    END;

  PROCEDURE write_xy(wx,wy : INTEGER; was : Str255);
    BEGIN
      write(#27,'Y',chr(wy+32), chr(wx+32),was);
    END;

  PROCEDURE hiwrite_xy(wx,wy : INTEGER; was : Str255);
    BEGIN
      write(#27,'p'); write_xy(wx,wy,was); write(#27,'q')
    END;

  PROCEDURE Rahmen(x1, y1, x2, y2 : INTEGER);
    { Doppelrahmen zeichnen und weiss ausmalen }
  VAR i : INTEGER;
    BEGIN
      Hide_Mouse;
      x2:=x2+2;
      x1:=(x1-1)*8+1;   x2:=(x2-1)*8;
      y1:=(y1-1)*16+6;  y2:=(y2-1)*16+12;
      Paint_Color(White); Paint_Rect(x1,y1,x2-x1,y2-y1);
      Line_Color(Black);
       Frame_Rect(x1,y1,x2-x1+1,y2-y1+1);
      FOR i:=3 TO 4 DO
       Frame_Rect(x1+i, y1+i, x2-x1-2*i+1, y2-y1-2*i+1);
      Show_Mouse
    END;

  { Die eigentliche Funktion, die jedoch obige Umgebung
    benoetigt.
  ---------------------------------------}

  FUNCTION flip(x, y : INTEGER; { Bildschirmkoordinaten }
                was  : str255;  { Menue-Eintraege       }
              VAR M_Keys : INTEGER) { Mausinfo          }
                         : INTEGER;
  TYPE Str40 = STRING[40];

  VAR  Feld              : ARRAY[1..23] OF str40;
       brstr             : Str40;
       Breite, anzahl,
       akt, last,
       xmin, xmax,
       ymin, ymax,
       xpos, ypos,
       p, i              : INTEGER;

    FUNCTION wx(x : INTEGER) : INTEGER;
      { Rechnet Bildschirmkoordinaten um in
        Pixelkoordinaten }
      BEGIN wx:=x*8-7 END;

    FUNCTION wy(y : INTEGER) : INTEGER;
      { s. wx }
      BEGIN wy:=y*16-1 END;

    PROCEDURE Aufspalten;
      { Menue-Eintraege auseinanderdividieren }
    VAR l : INTEGER;
        h : Str255;
      BEGIN
        Breite:=0; l:=0; Anzahl:=1;
        WHILE Pos('_',was)<>0 DO BEGIN
          { '_' als Trennungzeichen }
          l:=Pos('_',was)-1;
          IF l>Breite THEN Breite:=l;
          Feld[anzahl]:=Copy(was,1,l);
          Delete(was,1,l+1);
          anzahl:=anzahl+1
        END;
        Feld[anzahl]:=was;
         IF length(was)>Breite THEN Breite:=length(was);
        brstr:=Copy('                                        ',1,Breite);
          { ——— 40 Spaces oben ——— }
        FOR l:=1 TO anzahl DO BEGIN
         { Felder gleichlang machen }
```

```
         h:=Concat(' ',Feld[1],brstr);
         Feld[1]:=Copy(h,1,Breite+2)
       END;
       Breite:=Breite+2
    END;

  PROCEDURE Fuellen;    { Flipmenue fuellen          }
  VAR i : INTEGER;
    BEGIN
      Hide_Mouse;
       FOR i:=1 TO anzahl DO write_xy(x,y+i-1,Feld[i]);
      Show_Mouse;
    END;

  PROCEDURE Update;      { Flipmenue aktualisieren }
    BEGIN
      Hide_Mouse;
       IF last<>0 THEN write_xy(x,y+last-1,Feld[last]);
                                                { normal }
       IF akt<>0  THEN hiwrite_xy(x,y+akt-1,Feld[akt]);
                                                { invers }
      Show_Mouse
    END;

  BEGIN
     akt:=0; last:=0; ScreenBase:=LogBase;
    Aufspalten;
     xmin:=wx(x+1); xmax:=wx(x+Breite+1);
     ymin:=wy(y);    ymax:=wy(y+anzahl);
    Save_Screen;
     Rahmen(x, y, x+Breite, y+Anzahl+1);
    Fuellen;
    REPEAT
       Graf_MKState(xpos, ypos, M_Keys, i);
       last:=akt;
       IF ((xpos>=xmin) AND (xpos<=xmax)) AND
          ((ypos>=ymin) AND (ypos<=ymax))
        THEN akt:=(ypos-2-ymin) DIV 16 + 1
        ELSE akt:=0;
      IF last<>akt THEN Update;
       IF (M_Keys<>0) AND (akt=0) THEN write(#7)
    UNTIL (M_Keys<>0) AND (akt<>0);
       Restore_Screen;
      flip:=akt
    END;

  {——————————————————————————}

{ Beispiel - Hauptprogramm                                    }
BEGIN
   IF Init_GEM>=0 THEN BEGIN
     Init_Mouse;

     hilf:=Concat(                   { Hilfsstring fuer
                                       Menue-Eintraege:   }
       '  Datei laden_',              { Eintraege werden
                                       durch den Underscore }
       ' 'Datei speichern_',          { (_) getrennt.      }
       '  Datei neu erzeugen_',
       '  File umbenennen_',
       '  File loeschen_',
       '  Diskette pruefen_',
       '  Diskette formatieren  _',
       '  Freier Speicherplatz');

      test:=flip(15,3, hilf, mouse);

      test:=flip(38,1,
        '1_2_3_4_5_6_7_8_9_10_11_12_13_14_15_16_17_18_19_20
       _21_22_23', mouse);

      writeln; writeln;
      write('Eintrag ',test,' gewaehlt mit ');
      IF mouse=1 THEN writeln('linker Maustaste.');
      IF mouse=2 THEN writeln('rechter Maustaste.');
      IF mouse=3 THEN writeln('beiden Maustasten.');
     writeln('<RETURN>'); readln;
    Exit_GEM
  END
END.
```

Betr.: Programmier Praxis

Die "Progammier Praxis" ist ein fester Bestandteil der ST Computer geworden. Gedacht ist sie für all diejenigen, die ihren Rechner selbst programmieren, und dazu Tips und Anregungen gebrauchen können. Diese Rubrik kann aber nur dann bestehen, solange viele Leser (Sie eingeschlossen) sich daran beteilligen. Wir fordern Sie deshalb auf, Ihre Ideen in einen Umschlag zu stecken und auf dem schnellsten Weg zu uns zu schicken. Voraussetzungen gibt es praktisch keine: es muß lediglich interessant sein und so dokumentiert, daß auch andere Anwender daraus einen Nutzen ziehen können. Auch bei den Sprachen gibt es keinerlei Einschränkungen: PASCAL, C, BASIC; Assembler, Modula 2, APL, Fortan, LISP, Prolog, PEARL,... und sogar LOGO ist erwünscht.

Veröffentliche Programme werden natürlich angemessen honoriert. Bitte beachten Sie bei der Einsendung: Schicken Sie den Quelltext und das evtl. compilierte Programm, sowie die Dokumentation auf Papier und Diskette. Die Diskette wird zurückerstattet.

Einsendungen direkt an die Redaktion:
"MERLIN" Computer GmbH
ST Computer Redaktion
'Programmier Praxis'
Industriestraße 26
D-6236 Eschborn Tel.: 0 61 96 / 48 18 11



DIE SCHILKRÖTE GEHT FREMD

Turtle-Grafiken

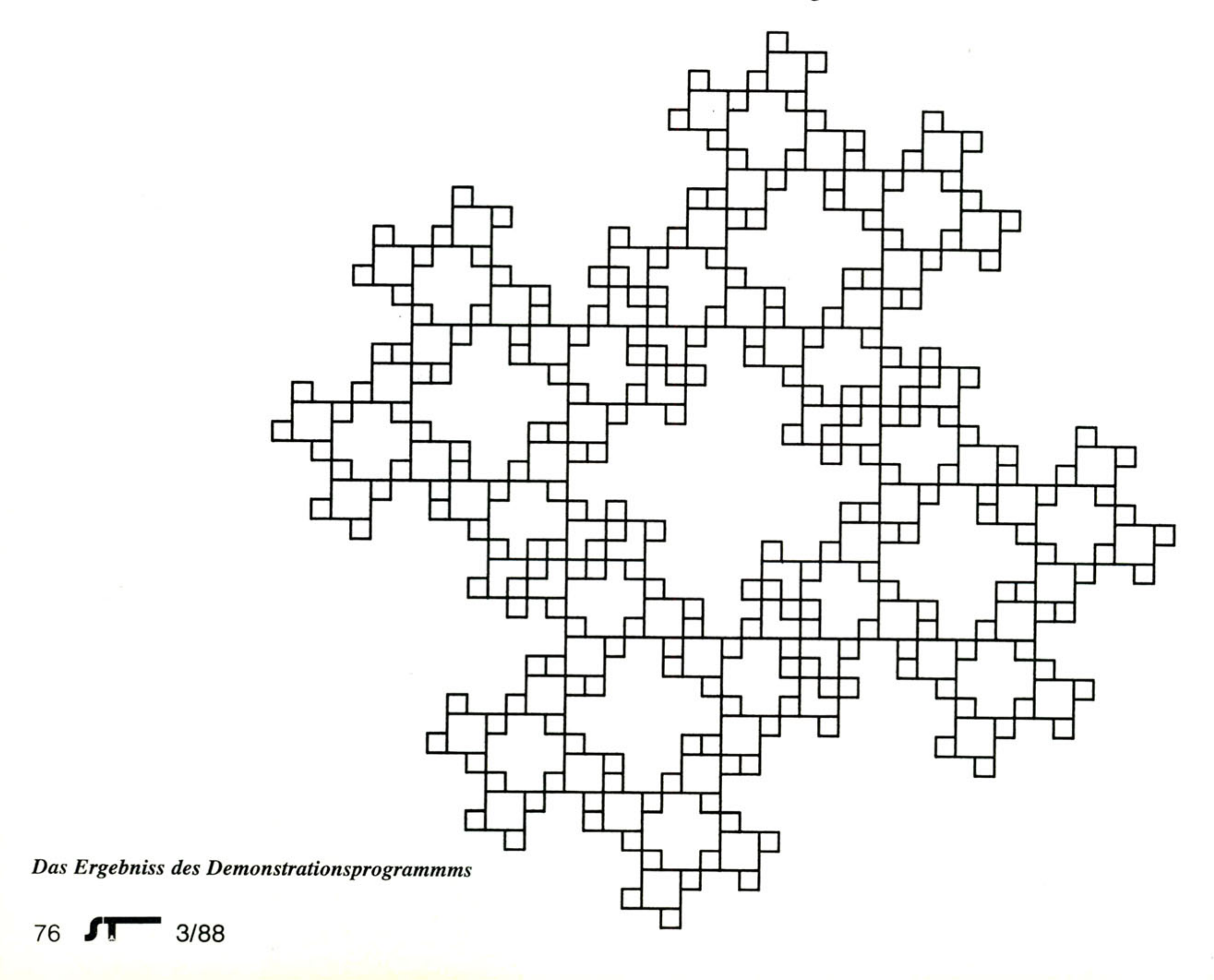
Turtle-Graphik ist nicht nur LOGO vorbehalten. Auch in anderen Hochsprachen wie Pascal lassen sich geeignete Routinen aufbauen.

Die hier vorgestellte Include-Datei läßt sich in jedes Programm einbinden. Allerdings sind die Routinen an GEM gebunden. TURTLE.INC stellt die vier Bewegungskommandos *vorwaerts*, *rueckwaerts*, *rechts* und *links* zur Verfügung. Zusätzlich sind zwei Routinen vorhanden, die die Turtle an einer bestimmte Stelle auf dem Bildschirm mit einer bestimmten Ausrichtung plaziert.

ST Pascal Plus 2.0 akzeptiert Konstanten-, Typen-, Variablen- und andere Mehrfachdeklarationen. Daher ist es möglich, alle notwendigen Routinen in einer Include-Datei zusammenzufassen. Sie darf aber erst nach Einfügen der GEM-Dateien eingefügt werden, da die Turtle-Datei Routinen des GEM benutzt.

Eine kurze Erläuterung. Mit der Routine init_turtle legt man einen beliebigen Startpunkt auf dem Bildschirm fest. Zusätzlich wird eine Grundausrichtung vorgegeben. turtle_home hat eine geringere Funktion. Sie entspricht init_turtle, setzt aber die Turtle in die Mitte mit Ausrichtung nach oben. vorwaerts bewegt die Turtle pixelweise vorwärts, das heißt in die durch den Winkel festgelegte Richtung. rueckwaerts verfährt analog, jedoch in die entgegengesetzte Richtung. Mit rechts und links kann man die Turtle-Richtung ändern. Hierbei ist immer ein Winkel von der Grundrichtung aus gemeint. Beispiel: Die Turtle bewegt sich in Richtung 90 Grad. Nun soll sie sich mit rechts(10) drehen. Die neue Richtung ist dann 80 Grad.
Bevor man eine der vier Bewegungsroutinen aufruft, sollte man die Turtle auf den Bildschirm mit init_turtle oder turtle_home setzen. Ansonsten gibt's nichts Besonderes zu beachten. Das kleine Beispielprogramm veranschaulicht Nutzungsmöglichkeiten der Turtle-Routinen.

Dietmar Rabich



Das Ergebniss des Demonstrationsprogrammms

TURTLE.INC

```
(* Include-Datei für Turtle-Routinen *)
(* Entwickelt mit ST Pascal Plus 2.0 *)
(* Dietmar Rabich, Dülmen            *)

type turtle_daten = record
                     pos_x,pos_y,winkel : short_integer;
                    end;

var turtle : turtle_daten; (* globale
                              Variablenvereinbarung *)

(* Turtle vorwärts bewegen *)
 procedure vorwaerts(n:short_integer);

  (* Umrechnung in Bogenma_ *)
  function bogen(x:real):real;

   const pi=3.14159265;

  begin
   bogen:=x*pi/180;
  end;

 begin
  with turtle do
   begin
      pos_x:=pos_x+short_round(cos(bogen(winkel))*n);
      pos_y:=pos_y-short_round(sin(bogen(winkel))*n);
     line_to(pos_x,pos_y);
   end;
 end;

(* Turtle rückwärts bewegen *)
 procedure rueckwaerts(n:short_integer);

 begin
  vorwaerts(-n);
 end;

(* Turtle rechts drehen *)
 procedure rechts(w:short_integer);

 begin
  with turtle do
   begin
     winkel:=winkel-w;
     while winkel<0 do
      winkel:=winkel+360;
   end;
 end;

(* Turtle links drehen *)
 procedure links(w:short_integer);

 begin
  with turtle do
   begin
     winkel:=winkel+w;
     while winkel>360 do
      winkel:=winkel-360;
   end;
 end;

(* Turtle Position und Winkel zuordnen *)
 procedure init_turtle(x,y,w:short_integer);

 begin
  with turtle do
   begin
    pos_x:=x;
    pos_y:=y;
    winkel:=w;
   end;
  rechts(0);
  links(0);
  move_to(x,y);
 end;

(* Turtle in die Mitte mit Ausrichtung nach oben *)
 procedure turtle_home;

 begin
   init_turtle(320,200,90);
 end;
```

```
(* Demonstrationsprogramm für die Include-Datei
   TURTLE. INC *)
(* Entwickelt mit ST Pascal Plus 2.0. Dietmar Rabich *)
program demo_turtle;

const {$I \INCLUDE\GEMCONST.INC }

type {$I \INCLUDE\GEMTYPE.INC }

{$I \INCLUDE\GEMSUBS.INC }

{$I \INCLUDE\TURTLE.INC } (* <- Hier wird die Turtle-
                                Datei eingelesen *)

 procedure quadrat(laenge:short_integer;dreh:boolean);

 var i : short_integer;

 begin
  if laenge>5 then
   for i:=1 to 4 do
    begin
      vorwaerts(laenge);
     if dreh then
       links(90)
      else
       rechts(90);
      quadrat(laenge div 2,not(dreh));
    end;
 end;

begin
 if init_gem>=0 then
  begin
   init_mouse;
   hide_mouse;
   clear_screen;
    init_turtle(384,264,90);
   quadrat(128,true);
   show_mouse;
   exit_gem;
   readln;
  end;
end.
```



ZEHNFINGER WIE DER WIND

Umdefinieren der Größer-/Kleiner-Taste

Wer seinen Atari im Einfinger-Adler-Suchsystem bedient und außerdem mit der Standardbelegung der Tastatur zufrieden ist, kann diesen Artikel getrost überschlagen.
Wer allerdings seinen Atari als Textsystem benutzt und sich wie ich schon des öfteren über die Lage der **<- bzw. >-Taste** geärgert hat, dem seien die folgenden Ausführungen wärmstens ans Herz gelegt.

Leider haben die Konstrukteure der Atari-Tastatur (das gilt auch für den neuen Mega Atari) nicht nachgedacht, als sie das Layout der deutschen Tastatur festlegten. Damit ziehen sie zwar gleich mit ihren Kollegen, den Entwicklern des Industriestandards, aber was kann sich der Anwender dafür kaufen? Alle, die auf ihrem Atari mehr oder weniger professionelle Textverarbeitung betreiben, werden es sicher schon festgestellt haben: Die '<'- bzw. '>'-Taste sitzt so bescheuert neben der linken Shift-Taste, daß die Texte am **<ende** meistens **<sO** ausehen !!
Vielleicht haben es einige auch geschafft, durch langwieriges Trainieren ihren kleinen Finger so zu verformen, daß sie das Problem nicht mehr haben; aber was machen die Ärmsten, wenn sie gezwungen werden, sich einer normalen Schreimaschinentastatur zu nähern ?

Gerade beim neuen Mega Atari, der garnicht typisch für PCs eine wirklich angenehme und qualitativ gute Tastatur hat, läßt obiger Umstand äußerste Traurigkeit aufkommen.
Kurz gesagt: Ich habe mich entschlossen, mit folgendem Programm der Sache ein Ende zu machen.

Zuerst hatte ich die Idee, aus der KG-Taste (diese Abkürzung stehe ab jetzt für die bewußte, störende Taste) eine dritte Shift-Taste zu machen. Allerdings ist mir das nicht gelungen. Für entsprechende Hinweise wäre ich sehr dankbar.
Als Alternative bot sich an, die KG-Taste völlig lahmzulegen und die entsprechenden Zeichen, ohne die der Programmierer natürlich auch nicht leben kann, woanders hinzulegen.
Dieser Weg zeigte sich als gangbar und wurde in vorliegendem (trivialen) Prográmmchen verwirklicht.

Effekt:
Die KG-Taste ist nach Aufruf des Programms völlig tot und sollte irgendwie mechanisch mit der linken Shift-Taste gekoppelt werden (Wer hat eine gute Idee ?, Holzkeil schnitzen, Plastillin, ...).
Nach einem zweiten Aufruf des Programms werden die alten Zustände wieder hergestellt.
Auf der Suche nach der Tastaturtabelle stieß ich auf die Funktion 16 des XBIOS, umgangssprachlich auch als **Keytbl()** bekannt. Dabei stellte sich heraus, daß der Atari nicht eine Tastaturtabelle, sondern deren drei hat. Jede ist genau 128 Byte lang und wird über den sogenannten **Scancode** der Tasten (s. Bild 1) addressiert. Der Scancode ist eine Art Numerierung aller Tasten.

Beispiel:
Sie wollen, daß die KG-Taste nur noch Null liefert. Also schauen Sie in der Tabelle (Bild 1) nach und sehen, daß die KG-Taste den Scancode 60 Hex hat. Deshalb müssen Sie in Position 60 Hex der shift- und unshift-Tabelle eine 0 eintragen.

Die erste Tabelle enthält die **normale** Belegung, die zweite die Belegung mit gedrückter **shift**-Taste und die dritte die bei aktiver **CapsLock**-Taste.
Mit der Funktion **Keytbl(u,s,c)** ist es nun möglich die Tabellen durch eigene zu ersetzen. Dazu ist der Funktion ein Zeiger auf die neue Tabelle zu übergeben. Wird als Parameter -1L übergeben, bleibt die jeweilige Tabelle unverändert.
Wenn man nun auch noch weiß, daß **Keytbl()** auch einen Wert zurückgibt - und zwar einen Zeiger auf drei weitere Zeiger, die, wen wundert's, auf die gerade gültigen Tastaturtabellen zeigen, dann ist der Rest ein Kinderspiel.
Zuerst besorgt man sich die alten Tabellen, indem man **Keytbl()** mit dreimal -1L als Parameter aufruft. Da sich die Originaltabellen im ROM befinden, muß man diejenigen, die man gern verändern möchte, ins RAM kopieren, dann abändern und dem Betriebssystem mit einem weiteren Aufruf von **Keytbl()** die neue Lage nahebringen.
Damit auch keiner an den neuen Tabellen fummelt, muß man das Programm mit **Ptermres(n,e)** terminieren, so daß es resident im Speicher bleibt. Wie man die Länge des Programms feststellt, hängt vom jeweiligen Compiler und der Programmiersprache ab. Für Megamax C können Sie eine mögliche Lösung im Listing bewundern. Bemerkenswert dabei ist, daß Megamax C der ersten globalen Variable die höchste Adresse zuordnet.
Die Problemlösung sollte sich auf alle Programmiersprachen übertragen lassen, die TOS-Aufrufe erlauben. Vielleicht hat jemand Lust, eine Version in Assembler zu machen und als Public Domain zur Verfügung zu stellen?

Thomas Weinstein

```c
/*
 * KEY.C
 *
 * Legt die Kleiner/Groesser-Taste lahm. Dies Taste hat
   nachher keine
 * Funktion mehr. Durch eine geeignete mechanische
   Verbindung koennen
 * nun die linke Shift-Taste und die Kleine/Groesser
   Taste verbunden
 * werden. Das erleichert das Schreiben im
   Zehnfingersystem ungemein.
 *
 * Ersatz: < auf Ziffernblock-( und
           > auf Ziffernblock-).
 *
 * Autor: Th. Weinstein
 * Datum: 09.09.87
 *
 * erstellt mit Megamax C
 *
 */

#include <osbind.h>

/*
 * Wird vom Xbios(16) Aufruf (Keytbl(a,b,c)) geliefert.
 */
typedef struct {
        char *unshift;      /* Zeiger auf Tastaturtabelle
                               ungeshifted */
        char *shift;        /* Zeiger auf Tastaturtabelle
                               geshifted   */
        char *capslock;     /* Zeiger auf Tastaturtabelle
                               CapsLock    */
} *KEYTBLS;

extern long   _base;     /* Beginn der Base Page */

long    plen; /* Zur Berechnung der Programmlaenge */

int     end_of_program;  /* Platz nur bis zu der
                            Adresse dieser     */
                  /* Variablen resident halten */

char    unshift[128];     /* Platz fuer neue Tabelle
                             ungeshifted      */
char    shift[128];       /* Platz fuer neue Tabelle
                             geshifted        */

/*
 * Hauptprogramm
 */
main()
{
        KEYTBLS   keytbls; /* Zeiger auf urspruengliche
                              Tabellen    */

        /*
         * Zuerst urspruengliche Belegung der Tastatur
           holen
         * Alle drei Parameter gleich -1L liefert alte
           Belegung
         * und aendert sonst nichts.
         */

        keytbls = (KEYTBLS) Keytbl(-1L,-1L,-1L);

        /*
         * Feststellen ob die Tabellen schon
           manipuliert wurden
         * (d.h. nicht mehr im ROM stehen)
         */
        if ((long) keytbls->unshift < 0xFC0000L)
        {
                Bioskeys(); /* alte Belegung wieder
                               herstellen */
                goto ENDE;
        }
        else
        {
                /*
                 * Tabellen für unshift und shift ins
                   RAM kopieren
                 */
                copy(keytbls->unshift,unshift,128);
                copy(keytbls->shift,shift,128);

                /*
                 * < und > - Taste lahmlegen
                 */
                unshift[0x60] = shift[0x60] = (char) 0;

                shift[0x63] = 0x3c;
                /* < auf Ziffernblock-( legen */
                shift[0x64] = 0x3e;
                /* > auf Ziffernblock-) legen */

                /*
                 * Neue Tabellen dem System bekannt
                   machen
                 */
                Keytbl(unshift,shift,-1L);

                /*
                 * Programmlaenge berechnen
                 */
                plen =   ((long) &end_of_program
                        - _base+ 1) - 2;

                /*
                 * Terminieren aber Speicher nicht
                   freigeben
                 */
                Ptermres(plen,0);
        }

ENDE: ;

}

/*
 * 'count' Bytes von 'from' nach 'to' kopieren. Keine
   Ueberpruefung auf
 * Ueberlappung der Bereiche.
 */
copy (from,to,count)
char *from, *to;
int   count;
{
        while (count--)
                *to++ = *from++;
}

/*
 * ENDE von KEY.C
 */
```

Die SCAN Codes der Atari Tastatur (alle Werte hexadezimal)

```
Funktionstasten:
     F1 - 3B   F2 - 3C   F3 - 3D   F4 - 3E   F5 - 3F   F6 - 40
     F7 - 41   F8 - 42   F9 - 43   F10- 44

Ziffernblock:
     (  - 63   )  - 64   /  - 65   *  - 66
     7  - 67   8  - 68   9  - 69   -  - 4A
     4  - 6A   5  - 6B   6  - 6C   +  - 4E
     1  - 6D   2  - 6E   3  - 6F   EN - 72
     0  - 70   .  - 71

Cursorblock:
     Help   - 62          Undo - 61
     Insert - 52 ↑ - 48   Clr  - 47
     ←      - 4B ↓ - 50   →    - 4D

Grosser Block:

Ziffern:
     1 - 02  2 - 03  3 - 04  4 - 05  5 - 06  6 - 07  7 - 08  8 - 09
     9 - 0A  0 - 0B

Buchstaben:
     A - 1E  B - 30  C - 2E  D - 20  E - 12  F - 21  G - 22  H - 23
     I - 17  J - 24  K - 25  L - 25  M - 32  N - 31  O - 18  P - 19
     Q - 10  R - 13  S - 1F  T - 14  U - 16  V - 2F  W - 11  X - 2D
     Y - 2C  Z - 15  ß - 0C  Ä - 28  Ö - 27  Ü - 1A

Sonderzeichen:
     < - 60  , - 33  . - 34  - - 35  ~ - 2B  + - 1B  ' - 0D  # - 29

Sonstige:
     ESC     - 01  TAB      - 0F  CNTRL     - 1D
     Shift-L - 2A Shift-R   - 36
     ALT     - 38  CapsLock - 3A  Leertaste - 39
```

ES IST EIN KREUZ MIT...

-Skalierung leicht erzeugt

Öfters werden Achsenkreuze für diverse Plots gebraucht. Ein kleines Unterprogramm in OMIKRON-Basic geschrieben bietet eine universelle Hilfe beim Programmieren solcher Skalierungen. Es sollte ohne größere Mühen in jedes andere Basic und mit einigen Mühen auch in andere Programmiersprachen übertragbar sein. Zwei Unterroutinen werden dazu zusätzlich benötigt:

Function Blanksweg$:
Beseitigt alle Blanks vor und nach einem String

Function Wandle$:
Erzeugt einen String mit gegebener Stellenanzahl vor und nach dem Komma.

Procedure Achse:
Eigentliche Skalierungsroutine einer Achse

Procedure Achskreuz:
Anwendung für ein Achsenkreuz Das Hauptprogramm zu Beginn zeigt einige Demoaufrufe.

Wichtig bei einer Skalierung ist, daß der Aufruf unabhängig für x und y erfolgen kann und die Unterscheidung erst in der Routine erfolgt. Man muß nur die Pixelkoordinaten 'Von' und 'Bis' in der Ordinate eingeben, die man benutzen will. Der Rest müßte aus den Kommentaren hervorgehen.

Im wesentlichen zur Problemlösung folgendes:

Unt! und Ob! sind die wahren x/y-Werte unten und oben, die nicht verändert werden, damit man nachher z.B. in ein solches Achsenkreuz Funktionswerte einzeichnen kann. Eine Skalierung muß nun die Achsenwerte so auftragen, daß sie in der üblichen 'Norm' bleiben, d.h. also Reihenfolge 1,2,3,4.. am Zahlenanfang bzw. 10, 15, 20, 25 oder 0, 2, 4, 6, 8... oder 25, 50, 75 usw., ohne aber die untere und obere Grenze zu ändern. Für einwandfreie Auswahl der unteren und oberen Grenze ist also der Benutzer verantwortlich.

Um das einwandfrei machen zu können, wird zuerst die Dimension der Größen gecheckt und die längste mögliche Stellenanzahl ermittelt, nach der sich die Skalierung richten muß, damit es keine krummen Zahlenwerte oder/und übereinandergeschriebene Werte gibt. Dazu wird die hochauflösende Norm mit 8*16-Pixelbuchstaben benutzt! Anschließend wird die Teilungsanzahl (die auf der 10er-Potenz beruht) in Einzelschritten solange geändert, bis eine sinnvolle und ausreichende Anzahl erreicht ist, so daß die Werte auf der Skala in ausreichendem Abstand zueinander stehen. Die feine Teilung erfolgt dann in 10* feineren Schritten (Huepf! im Ggs. zu Teil!).

Demo für verschiedene y-Skalierungen bei gleicher Länge

0.000	0.00	0.0	0.00	0.0	0	0	0	0	0	0	0
0.002	0.02	0.1	0.25	0.5	1	2	5	5	10	20	20
0.004	0.04	0.2	0.50	1.0	2	4	10	10	20	40	40
0.006	0.06	0.3	0.75	1.5	3	6	15	15	30	60	60
0.008	0.08	0.4	1.00	2.0	4	8	20	20	40	80	80
0.010	0.10	0.5	1.25	2.5	5	10	25	25	50	100	100
0.012	0.12	0.6	1.50	3.0	6	12	30	30	60	120	120
0.014	0.14	0.7	1.75	3.5	7	14	35	35	70	140	140
0.016	0.16	0.8	2.00	4.0	8	16	40	40	80		160
0.018	0.18		2.25	4.5	9	18		45	90		180
			2.50	5.0	10	20		50	100		200
				5.5	11	22		55			
				6.0	12			60			
								65			



Für die eigentliche Plotroutine (For P!= usw.) habe ich zur Geschwindigkeitssteigerung Hilfsgrößen vereinbart, da sie sonst doch arg langsam wird. Wichtig bei der Plotroutine ist, daß die Werte, die auf die kleinste Skaleneinheit normiert wurden (teilbar ohne Rest...), sich aber immer auf die unterste WIRKLICHE Grenze (Unt!) beziehen, da ansonsten die Skala schön, aber nicht richtig wäre!. Im Programm prüft man dann auf Teilbarkeit ohne Rest auf die großen Distanzen für die Werteschreibung.

Das Problem am ganzen Programm ist, daß mit Benutzung des Logarithmus immer kleinste Rechenreste hinzukommen. Daher darf man auch nicht auf Modulrest 0 genau prüfen, sondern muß immer auf <= einen absoluten Minimalbetrag checken, oder die Werte entsprechend vorher verringern oder vergrößern - um einen kleinen Minimalbetrag, der immer kleiner als 1 Promille bleiben muß (Auflösung des Schirms). Dieses sind die kleinen 1E-5, die hier und dort im Programm auftauchen. Ansonsten kommt es dann und wann zu unerwünschten Skalierungen, die nicht 'schön' sind. Das Programm in dieser Form ist durch Anklicken auf OM-BAS-R.PRG lauffähig und die Routine aus dem LST-File nach Löschen des Hauptprogrammes und entsprechendem Umnumerieren mergefähig. Wichtig ist, daß Achse die Funktionen Blanksweg$ und Wandle$ benötigt, die es aufruft. Mehr zur Funktionsweise zu sagen, würde Bände füllen. Ein wenig Vertiefung in das eigentlich gut kommentierte Programm dürfte auch Klarheit schaffen.

Jost Jahn

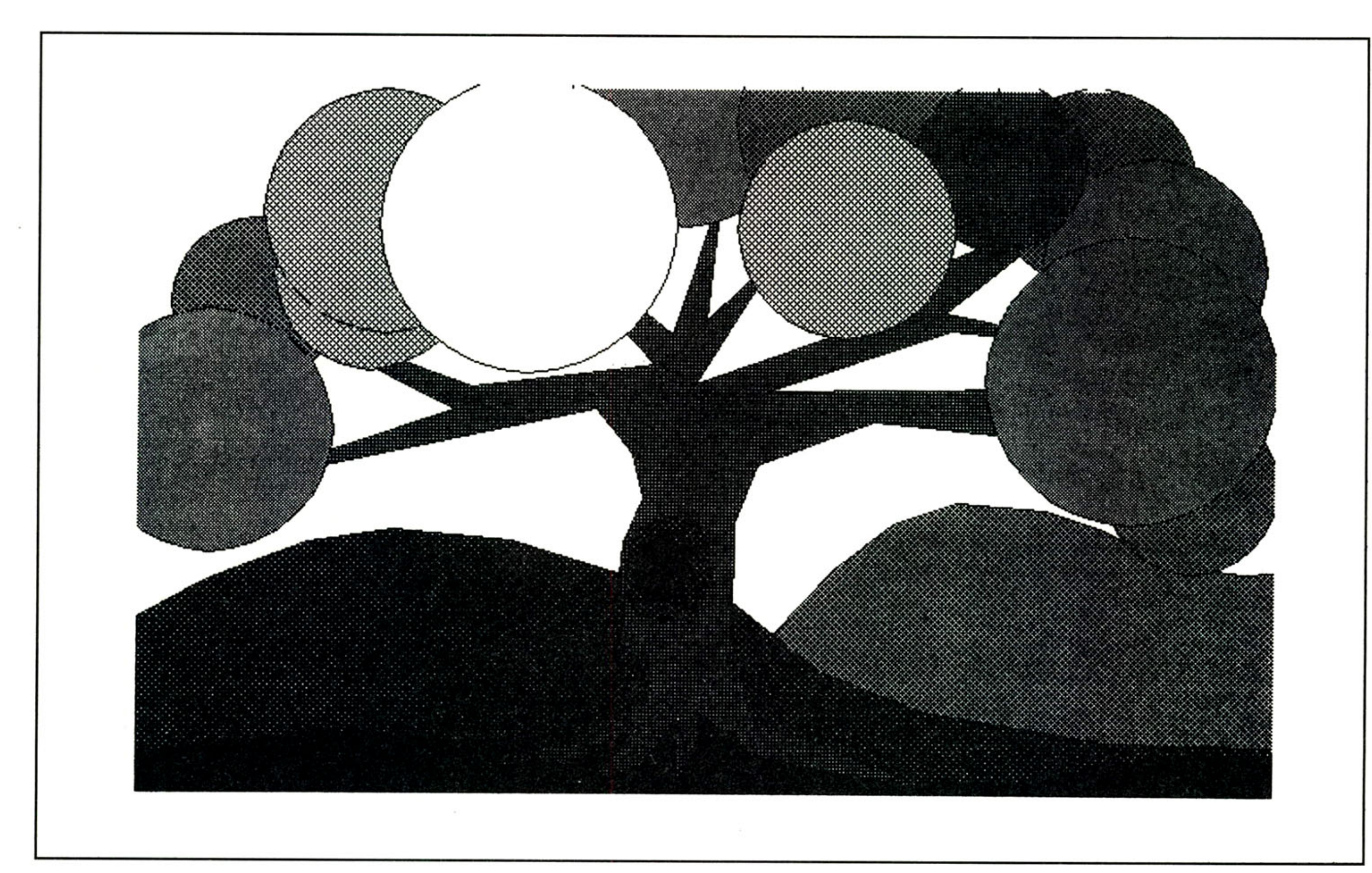
```
  ' -=> Demoprogramm für die Skalen (hohe Auflösung)
 CLS ' Schirm löschen
  TEXT 50,28,"Demo reines Achsenkreuz mit einfachem
               Raster" ' 1.Demo
   Achskreuz(50,590,370,70,-30,80,20,60,0,1)
  ' Einfaches Achskreuz
 WAIT 10 ' 10 Sekunden zeigen
 CLS      ' Schirm löschen
  TEXT 50,28,"Demo Achsenkreuzkasten mit doppeltem
               Raster" ' 2.Demo
   Achskreuz(50,590,370,70,-30,80,20,60,1,2)
  ' Einfaches Achskreuz
 WAIT 10 ' 10 Sekunden zeigen
 CLS      ' Schirm löschen
  TEXT 50,28,"Demo für verschiedene x-Skalierungen bei
               gleicher Länge" ' 3.Demo
 FOR I=40 TO 370 STEP 30 ' Skalenschleife x
    Achse(40,600,I,I,0,(I/190)^5.3,-2,0) ' Skala in x
 NEXT I
 WAIT 10 ' 10 Sekunden zeigen
 CLS      ' Schirm löschen
  TEXT 50,28,"Demo für verschiedene y-Skalierungen bei
               gleicher Länge" ' 4.Demo
 FOR I=70 TO 620 STEP 50 ' Skalenschleife y
    Achse(50,370,I,I,0,(I/177)^4.3,2,0) ' Skala in y
 NEXT I
 WAIT 10 ' 10 Sekunden zeigen
 CLS      ' Schirm löschen
  TEXT 50,28,"Demo für verschiedene Skalierung und
               Raster" ' 5. Demo
  Achse(60,580,366,60,-30,30,-2,2) ' x-Achse unten
  Achse(60,580,60,366,1210,1250,-1,0) ' x-Achse oben
  Achse(60,366,60,580,-.05,-.01,2,3) ' y-Achse links
  Achse(60,366,580,60,35,40,1,0) ' y-Achse rechts
 WAIT 10 ' 10 Sekunden zeigen
 CLS      ' Schirm löschen
  TEXT 50,28,"Demo für gleiche Skalierung und
               Richtungsumkehr mit Mischraster"
  Achse(550,60,360,60,-20,60,-2,3) ' x-Achse untèn
  Achse(60,550,60,360,-20,60,-1,2) ' x-Achse oben
  Achse(360,60,60,550,-.4,.1,2,4)  ' y-Achse links
  Achse(60,360,550,60,-.4,.1,1,3)  ' y-Achse rechts
 WAIT 60 ' 10 Sekunden zeigen
 END
  ' #####################
  ' # Achsenkreuzkasten #
  ' #####################
   DEF PROC Achskreuz(Links,Rechts,Unten,Oben,Xmin!,
            Xmax!,Ymin!,Ymax!,Wie,Raster)
     ' Links - Pixelwert linke  Kante
       Rechts - Pixelwert rechte Kante
     ' Unten - Pixelwert untere Kante
       Oben   - Pixelwert obere  Kante
     ' Xmin!, Xmax! - Skalenwerte x
     ' Ymin!, Ymax! - Skalenwerte y
     ' Wie: 0 Nur Achsenkreuz, 1 Achskasten
     ' Raster: Rasterbreite in 4er-Pixel (0 kein Raster)
      Achse(Links,Rechts,Unten,Oben,Xmin!,Xmax!,-2,Raster)
    IF Wie=1 THEN Achse(Links,Rechts,Oben,Unten,Xmin!,
                    Xmax!,-1,0)
      Achse(Unten,Oben,Links,Rechts,Ymin!,Ymax!,2,Raster)
    IF Wie=1 THEN Achse(Unten,Oben,Rechts,Links,Ymin!,
                    Ymax!,1,0)
 RETURN
  ' #####################
  ' # Skalierung Achsen #
  ' #####################
   DEF PROC Achse(Von,Bis,Wo,Hier,Unt!,Ob!,Was,Rast)
     ' Von - Startpunkt der Achse,
       Bis - Zielpunkt der Achse
     ' Wo - Andere Ordinate Achse selbst,
       Hier - Andere Ordinate für Raster
     ' Unt! - Realer unterster Wert,
       Ob! - Realer öberster Wert
     ' Was - Achsrichtung (>0 y-Achse, <0 x-Achse,
       1 dann rechts, 2 links,
     ' -2 unterhalb und -1 oberhalb der Achse)
     ' Rast - Raster zeichnen (0 nein, x ja im Abstand
       4er-Punkte)
     ' Umkehr der Werterichtung: Bis und Von vertauschen
     LOCAL Unten_Dim,Oben_Dim,Stellen,P!,Da,K,Huepf!,
```

```
      Wieoft,St ' Lokale Variablen
   LOCAL Stel,Dist!,Teil!,Teil_Dim,Teil,Unten!,Oben!,
      Teilmax ' Lokale Variablen
   LOCAL Hilf1,Hilf1!,Hilf2,Hilf3,Hilf4,Hilf5,Hilf6,
      Merken '   Zur Beschleunigung
  Rast= ABS(Rast) ' Nur positives Raster
  Dist!= ABS(Ob!-Unt!) ' Differenz Skalenenden
  Ob!=Ob!+Dist!*1E-5   ' Oben bi_chen mehr
  Unt!=Unt!-Dist!*1E-5 ' Unten bi_chen weniger
  Unten!=Unt!:Oben!=Ob!' Skalenenden
  Unten_Dim= LOG(10, ABS(Unten!)+1E-5) ' Dimension
                                        untere Grenze
  Oben_Dim= LOG(10, ABS(Oben!)+1E-5) ' Dimension
                                        obere Grenze
  St= MAX(Oben_Dim,Unten_Dim) ' Optimale Stellenanzahl
  ' — Ermittlung der maximalen Stellenzahl
  Stel= ABS(St)+2-2*(St<=1)-(Un!<0) ' Maximale
                                        Stellenzahl merken
 IF Was>0 THEN ' y-Achse
   DRAW Wo,Von TO Wo,Bis ' y-Achse zeichnen
   Hoehe= ABS(Bis-Von) ' Pixelhöhe der Achse
   Teilmax= INT(Hoehe/22) ' max. Zahleinteilung
 ELSE '                   x-Achse
   DRAW Von,Wo TO Bis,Wo ' x-Achse zeichnen
   Breite= ABS(Bis-Von)  ' Pixellänge der Achse
   Teilmax= INT(Breite/10/Stel)' max. Zahleinteilung
 ENDIF
  ' — Ermittlung der groben Teilung (Zahlenwerte)
 Stellen=St ' Zuweisung der Stellen
 REPEAT '     Bei engen Werten solange ...
   Stellen=Stellen-1 ' ... erniedrigen, bis ...
   Huepf!=10^Stellen ' ... Schrittweite OK.
   Oben!=( INT(Ob!/Huepf!))*(Huepf!) ' Grenze oben
                                        drunter setzen
   Unten!=( INT(Unt!/Huepf!)+1)*(Huepf!)' Grenze
                                        unten drüber setzen
  UNTIL ABS((Oben!-Unten!)/(Huepf!))>Teilmax
  ' Feinstriche md. sooft wie Teilung
 Teil!=Huepf!*10 ' Variable grobe Teilung
 Teil_Dim= INT( LOG(10,Teil!)) ' Dimension des Teils
 Teil=(Oben!-Unten!)/Teil!      ' Ganzanteil
 Merken=0                ' Merkvariable für .25 Teilung
 IF(Teil*5)<=Teilmax THEN Teil!=Teil!/5:Teil=Teil*5
 ' Skalenteile zu wenig
 IF(Teil*2)<=Teilmax THEN Teil!=Teil!/2:Teil=Teil*2
 ' Skalenteile zu wenig
 IF(Teil*5)<=Teilmax THEN Teil!=Teil!/5:Teil=Teil*5
 ' Immer noch zu eng
 IF(Teil*2)<=Teilmax THEN Teil!=Teil!/2:Teil=Teil*2:
 Merken=1 ' .25-Teilung!
 ' — Ermittlung der feinen Teilung (Striche)
 Huepf!=Teil!/10
 Oben!=( INT(Ob!/Huepf!))*(Huepf!)' Grenze oben
                                        drunter setzen
 Unten!=( INT(Unt!/Huepf!)+1)*(Huepf!)' Grenze unten
                                        drüber setzen
 Dist!=Oben!-Unten!        ' Differenz Skalenenden
 Wieoft=Dist!/Huepf!       ' ..neue Anzahl der Schritte
 Vorher=Stel+2             ' Stellen vor dem Komma
 Grenze!=Huepf!/10         ' Limitdefinition
 Hilf1!= INT( LOG(10,Teil!)+1D-8)' Hilfsgrö_e
 Nachher= MAX(0,-Hilf1!)+Merken ' Nachkommastellen
 Hilf1!=(Bis-Von)/(Ob!-Unt!)     ' 1. Hilfsgrö_e zur
                                        Beschleunigung
 Hilf1=(1.5-Was)*4' 2. Hilfsgrö_e zur Beschleunigung
 Hilf2=Rast*3* SGN(Hier-Wo)      ' 3. Hilfsgrö_e zur
                                        Beschleunigung
 Hilf3=(1.5+Was)*4' 4. Hilfsgrö_e zur Beschleunigung
 Hilf4=10*(Was=1):Hilf5=8*(Was=2)' 5.+6. Hilfsgrö_e
 Hilf6=-24*(Was=-2)+9*(Was=-1)   ' 7. Hilfsgrö_e
 FOR P!=Unten! TO Oben!+Dist!*1E-5 STEP Huepf!
                                 ' Schleife
  P$=FN Wandle$(P!,Vorher,Nachher)' String bilden
  P$=FN Blanksweg$(P$)           ' Blanks weg
   Da=Von+(P!-Unt!)*Hilf1!' Ort ermitteln(wahrer Ort!)
   Pmodul!= ABS( INT((P!+Grenze!)/Teil!)*Teil!-P!)
   ' Wertentscheidungsvariable
  IF Was>0 THEN                  ' y-Achse
     DRAW Wo,Da TO Wo+Hilf1,Da   ' Zeichnen des
                                   feinen Striches
     IF Pmodul!<Grenze! THEN     ' Platz für Text
        DRAW Wo,Da TO Wo+Hilf1*2.5,Da ' Grober Strich
        TEXT Wo-Hilf4+(1+ LEN(P$))*Hilf5,Da+7,P$
                                       ' Zahl schreiben
        IF Rast>0 THEN
           FOR K=Wo TO Hier STEP Hilf2: DRAW K,Da: NEXT
           K ' Raster
        ENDIF
     ENDIF
    ELSE '                           x-Achse
      DRAW Da,Wo TO Da,Wo-Hilf3' zeichnen
        des feinen Striches
      IF Pmodul!<Grenze! THEN '  Platz für Text
        DRAW Da,Wo TO Da,Wo-Hilf3*2.5 ' Grober Strich
        TEXT Da- LEN(P$)*4,Wo+Hilf6,P$' Zahl schreiben
       IF Rast>0 THEN
          FOR K=Wo TO Hier STEP Hilf2: DRAW Da,K: NEXT
          K ' Raster
       ENDIF
      ENDIF
    ENDIF
  NEXT P!
 RETURN
  ' ##################################
 ' # Löscht Blanks vorne und hinten #
  ' ##################################
 DEF FN Blanksweg$(Was$)
   LOCAL Vorne,Hinten'      Lokale Variablen
   FOR Vorne=1 TO LEN(Was$)' 1. Buchstaben ungleich
                              Blank
     IF MID$(Was$,Vorne,1)<>" " THEN EXIT
   NEXT Vorne
   FOR Hinten= LEN(Was$) TO 1 STEP -1' Letzten
                              Buchstaben ungleich
     IF MID$(Was$,Hinten,1)<>" " THEN EXIT
   NEXT Hinten
   IF Vorne<1 THEN Vorne=1'   Probleme korrigieren
    IF(Hinten-Vorne)<1 THEN Hinten=Vorne+1'Bei Problemen
                              korrigieren
  RETURN MID$(Was$,Vorne,Hinten-Vorne+1)'   Blankfrei
                              zurückgeben
  ' #########################################
 ' # String bilden mit Vor und Nach-Stellen #
  ' #########################################
 DEF FN Wandle$(Wert!,Vor,Nach)
   LOCAL Total,Wie$,Wxu$'      Lokale Variablen
   IF Nach<0 THEN Nach=0'      Nach Minimum 0
   IF Vor<1 THEN Vor=1'        Vor Minimum 1
    Total=Vor+Nach+1+(Nach=0)'  Totallänge
   Wie$="#"*Vor'               Vorstellen
    IF Nach>0 THEN Wie$=Wie$+"."+"#"*Nach'Nachstellen
   USING Wie$'                       So gebrauchen
   Wxu$= STR$(Wert!)'                Wert zuweisen
   USING '                   Wieder Normalgebrauch
 RETURN Wxu$'                und Wert zurück..
```



ALGORITHMEN & DATENSTRUKTURE IN PASCAL

Teil 5 : AVL-Bäume



Heute möchte ich Ihnen eine Verfeinerung der Baumstruktur vom letzten Mal vorstellen, die AVL-Bäume. Benannt nach ihren Erfindern, den Herren **A**delson-**V**elskii und **L**andis, beheben Sie ein großes Manko, welches botanische und informatische Bäume immer besitzen: 'Sie wachsen wild !'

Sinn und Nutzen der AVL-Bäume

Dazu betrachten wir einige mögliche Bäume, die durch das Einfügen von sieben Elementen in die Baumstruktur vom letzten Mal entstehen können.

Wie Sie in Abbildung 5a leicht erkennen, sehen, in Abhängigkeit von der Eingabereihenfolge, die resultierenden Bäume sehr unterschiedlich aus. Bei der linken Eingabesequenz sehen Sie, daß der Baum zu einer linearen Liste wird. Dabei geht der

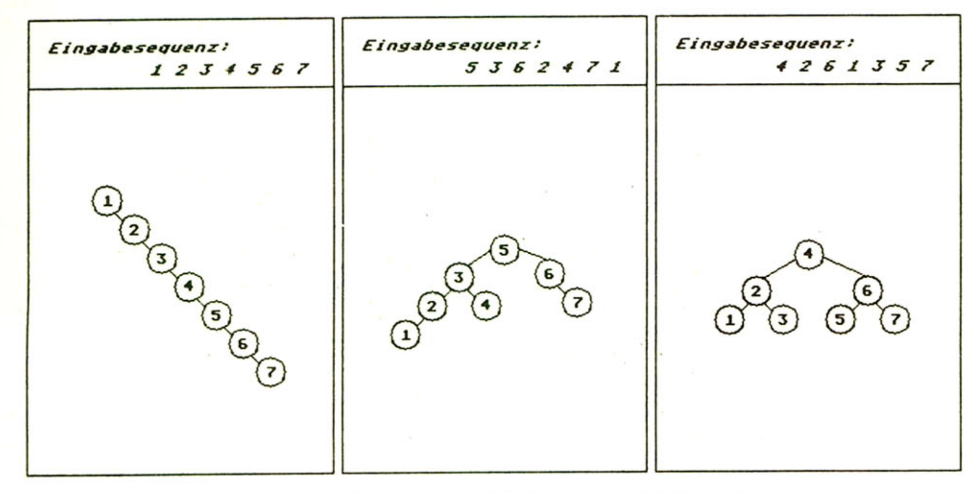


Bild 5a: Von degenerierten, vollständig und nur ein 'bißchen' ausgeglichenen Bäumen

große Vorteil der Suchbäume, die kurzen Suchwege, verloren. Deshalb nennt man diesen Baum und andere Bäume, denen man die Baumstruktur nicht mehr ansehen kann, **degenerierte Bäume**.

Die Aufgabenstellung ist es nun, eine Möglichkeit zu finden, nach Ein- und Ausfügen in Bäumen durch Ausgleichsmechanismen wieder kurze Suchwege zu erreichen, also die Degenerierung von Bäumen zu verhindern. Dabei ist der rechts in Abbildung 5a dargestellte Idealzustand in der Regel nur mit großem Aufwand herzustellen. Deshalb benötigen wir zunächst eine etwas schwächere Definition der Ausgeglichenheit, die effektive Algorithmen erlaubt. Dazu betrachten wir eine Definition für einen **ausgeglichenen** Baum, die auf die schon erwähnten Herren A.V.L. zurückgeht:

Ein Baum wird genau dann als 'ausgeglichen' bezeichnet, wenn für alle seine Knoten gilt, daß sich die Höhen des linken und des rechten Teilbaumes um nicht mehr als eins unterscheiden.

Betrachtet man, mit Augenmerk auf diese Definition, die Bäume der Abbildung 5a, ist klar:

Der rechts dargestellte Idealzustand, auch als **vollständig ausgeglichen** bezeichnet, ist ein AVL-Baum. Aber auch schon der mittlere Baum gibt sich nach kurzem Hinschauen als AVL-Baum zu erkennen. Gerade dieser kleine Spielraum zwischen **ausgeglichen** und **vollständig ausgeglichen** macht die Anzahl der nach Ein- und Ausfügen in Bäumen notwendigen Ausgleichsvorgänge beim AVL-Verfahren vertretbar.

Balance-Faktoren

Bevor wir uns an die einzelnen Mechanismen machen, möchte ich kurz auf die Höhenermittlung der Teilbäume eingehen. Wie Sie sicherlich leicht einsehen, ist es völlig unwirtschaftlich, nach jeder Operation, die die Baumstruktur verändert, die Länge sämtlicher Teilbäume zu ermitteln. Stattdessen bietet es sich an, in die Baumstruktur, speziell in jedem Baumelement, einen zusätzlichen Merker einzubauen, der die Höhendifferenz des rechten und des linken Teilbaums verzeichnet. Da wir es mit AVL-Bäumen zu tun haben, hat dieser Merker nur drei mögliche Werte:

-1 : Der linke Teilbaum ist um eins höher.
0 : Beide Teilbäume sind gleichhoch.
+1 : Der rechte Teilbaum ist um eins höher.

Diesen Merker, **bal** genannt, werde ich zusätzlich in die **b_element**-Definition der letzten Folge aufnehmen. Zusammen mit dem an unsere heutige Anwendung - eine Adreßdatei - angelehnten Schlüssel und der Datentypdefinition, ergeben sich die Typen des Listings 5a.

An den Operationen der letzten Folge bewirken diese Vereinbarungen folgende Änderungen:

Einfügen

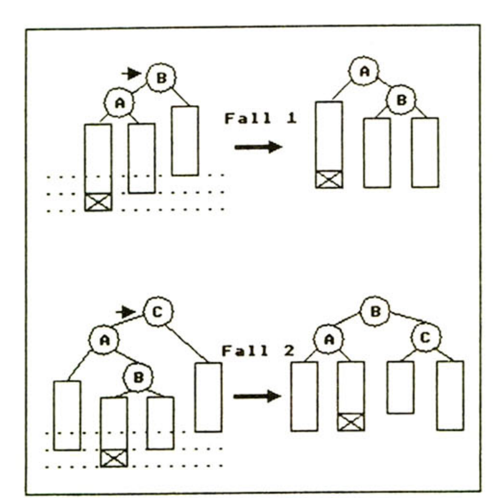
Das Einfügen in AVL-Bäume unterscheidet sich zunächst in keiner Weise von dem Einfügen in einfache binäre Bäume. Auch hier muß, über die rekursive Baumsuche, der Suchpfad bis zur Einfügestelle ermittelt werden.

Nach dem Einfügen ist aber zusätzlich die Höhenänderung zu berücksichtigen. Dazu wird in die Parameterliste der Funktion **put_in** (rekursive Subprozedur von **insert**) zusätzlich ein bool'scher Parameter **h** aufgenommen, der wahr wird, wenn eine Höhenänderung stattgefunden hat; andernfalls wird er wahr.

Anmerkung: Nach dem Einfügen ist **h** zunächst immer wahr, weil auf jeden Fall eine Balancefaktoränderung des Vater knotens der Einfügestelle auftritt.

Der rekursive Charakter von put_in kommt uns nun sehr zustatten. Wir müssen nur nach dem rekursiven Aufruf von put_in eine Abfrage auf eine Höhenänderung starten. Hat eine Höhenänderung stattgefunden, bestehen prinzipiell drei Möglichkeiten:

1. Die Höhenänderung eines Teilbaumes gleicht ein zuvor bestehendes Ungl eichgewicht des AVL-Baumes aus. In diesem Fall wird der Balancefaktor '0', und in den höher liegenden Knoten findet keine Höhenveränderung mehr statt; h ist also auf 'false' zu setzen.
2. Die Höhenänderung trifft bei einem Knoten mit gleich hohen Teilbäumen ein. Hier findet eine Balancefaktor-



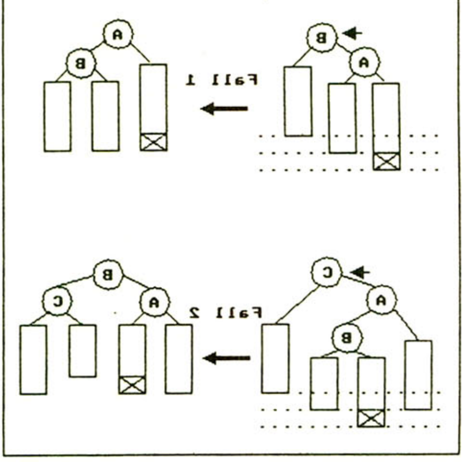


Bild 5b: Vier Ausgleichssituationen beim Einfügen und ihre Symmetrie

änderung im Rahmen der AVL-Eigenschaft statt. Der Balancefaktor wechselt von '0' zu '-1', oder '+1', jenachdem, in welchem Teilbaum eingefügt wurde. Diese Änderung bewirkt nun aber auch eine Änderung der Teilbaumhöhe des nächsthöheren Knotens. **h** ist also auf dem Wert 'true' zu belassen.
3. Die Höhenänderung verletzt die AVL-Eigenschaft. In diesem Fall sind Ausgleichsoperationen auszuführen.

Nach gründlicher Überlegung kommt man zu dem Ergebnis, daß sich, bis auf Symmetrie, nur zwei mögliche Ausgleichssituationen ergeben können.
Diese beiden und die symmetrischen Situationen sind in der Abbildung 5b dargestellt. Entsprechend den Bewegungen der an den Rotationen beteiligten Knoten A-C nennt man:

```
Fall 1a eine L-Rotation,
Fall 1b eine R-Rotation,
Fall 2a eine LR-Rotation,
Fall 2b eine RL-Rotation,
```

wobei L für Links- und R für Rechts-Rotationen steht.
Die gestrichelten Linien verdeutlichen sehr schön, wie die Teilbäume, die die AVL-Eigenschaft verletzen, bei den entsprechenden Rotationen angehoben werden.
Im Listing findet man die Zeigerrotationen für L in den Zeilen 78-81, für LR in 85-98, für R in 120-123 und für RL in 127-140.
Bemerkenswert ist dabei, daß durch lediglich eine (!) Rotation die AVL-Eigenschaft wieder hergestellt wird. Bei der AVL-Löschoperation werden wir damit im Regelfall nicht auskommen.
Doch zuvor noch ein kleines Beispiel zum Einfügen. Betrachten Sie die Abbildung 5c. Hier ist der AVL-Baum dargestellt, der durch Einfügen der Elemente 9, 1, 3, 5, 4 und 10 entsteht. Die Pfeile markieren dabei die Knotenelemente mit nach dem Einfügen verletzter AVL-Eigenschaft. Die Kreise deuten die Rotationen an und dürften somit die R-L-Namensgebung verdeutlichen.
Eine kleine technische Änderung möchte ich auch nicht unerwähnt lassen. Um das Überlaufen des Pascal-Stacks bei der **new**-Operation zu verhindern, wird in Zeile 155 berechnet, ob der benötigte Speicherplatz für das neu hinzukommende Element ausreicht. Dabei kommen die beiden Pascal+-Operationen **memavail** und **sizeof** zum Einsatz. **memavail** berechnet den verfügbaren Speicher in Worten; **sizeof** das Format des übergebenen Typs in Byte.
Ist der Speicher (**store**) ausreichend, wird das Element eingefügt. Der bool'sche Rückgabeparameter von **insert** bekommt den Erfolg dieser Aktion zugewiesen.

Löschen

Auch die Löschoperation gestaltet sich nicht prinzipiell anders, als in der letzten Folge. Hier erfolgt ebenfalls zunächst die rekursive Baumsuche (in **delete**).
Zusätzlich zur letzten Folge ist hier, wie schon bei insert, eine bool'sche Hilfsvariable h mit in die Parameterliste aufgenommen worden. Diese zeigt jetzt allerdings eine Höhenminderung an.
Das Ausfügen des Knotens selber gliedert sich wieder in drei Fälle:

1. Kein rechter Nachfolger
2. Kein linker Nachfolger
3. Zwei echte Nachfolger

Für den nicht-trivialen letzten Fall ist diesmal, abweichend von der letzten Folge, eine rekursive Ausfügeprozedur (del) angegeben. Die Zeigerrotation (**balance1** und **balance2**), deren Aussehen den Rotationen beim Einfügen ähnelt, ist diesmal, um redundanten Kode zu vermeiden, als Prozedur formuliert. Ansonsten realisieren sie die gleichen Aufgaben wie die Rotationen in insert. Der entscheidende Unterschied ist, daß beim Löschen eventuell mehr als eine Ausgleichsoperation durchgeführt werden muß.
Eine weitere Änderung gegenüber dem letzten Mal ist die Behandlung eines oder mehrerer Elemente, die ausgefügt werden sollen.
Wurden sie beim letzten Mal lediglich aus dem Baum ausgeklinkt, so wird diesmal, über die Variable **del_list** in der Parameterliste von **remove**, ein Zeiger auf die ausgefügte Elementliste übergeben. Dadurch besteht die Möglichkeit, entweder durch die aufrufende Programmstelle, die Speicherfreigabe der Elementnachfolgerliste zu betreiben, oder mit dieser Liste eine Sicherheitsabfrage zu durchlaufen, in der einzelne oder alle Listenbestandteile wieder in den Baum eingefügt werden können. Wie diese letzte Möglichkeit handzuhaben ist, wird sich in unserer heutigen Anwendung zeigen.

Andere AVL-Operationen

Die anderen AVL-Operationen unterscheiden sich nicht wesentlich von den Operationen der letzten Folge. Um die Fehleranfälligkeit der Dateioperationen zu senken, sind diese jedoch mit einem I/O-Laufzeitcheck versehen worden. Dazu sind im Kopf des Listing 5b die beiden externen Pascal+ Operationen **io_check** und **io_result** vereinbart worden. Mit io_check(false) wird vor Beginn des Datenzugriffs auf die Datei der IO-Check abgeschaltet. Dies bewirkt, daß sich Pascal+ nicht bei jedem kleinen IO-Fehler verabschiedet und im Anschluß an einen Dateizugriff die Fehlerkennung mit io_result erfragt werden kann.
Entsprechend der Möglichkeit eines fehlerhaften Dateizugriffs wurden die beiden dazu vorgesehenen AVL-Operationen, **load** und **save**, als Funktionen mit bool'schem Rückgabewert deklariert.

geht weiter...

Bild 5c: Einfügen an einem Beispiel

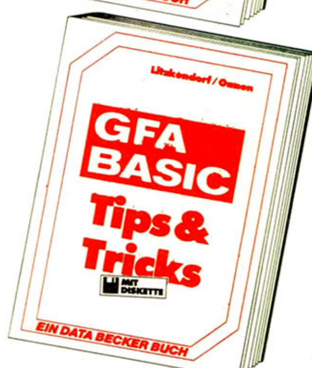
ATARI ST BUCHHITS!



Auspacken und gleich loslegen. Atari ST für Einsteiger macht's möglich. Vom Aufstellen und Anschließen über die Arbeit mit dem GEM-Desktop bis hin zum ST-BASIC – mit diesem Buch haben Sie die optimale Einführung zu Ihrem neuen Rechner. So ist der Erfolg bei Ihrer späteren Arbeit sozusagen schon vorprogrammiert.
ATARI ST für Einsteiger
248 Seiten, DM 29,–



Das meistverkaufte Buch zu GFA-BASIC – und das nicht ohne Grund. Denn hier lernen Sie den kompletten Befehlssatz des GFA-BASIC Version 2.0 und den GFA-Compiler anhand zahlreicher Beispielprogramme kennen. Ganz ohne die sonst übliche, nackte Befehlsübersicht. Das große GFA-BASIC-Buch – natürlich mit einem ausführlichen Einsteigerteil.
Das große GFA-BASIC-Buch
Hardcover, 574 Seiten
DM 49,–



Profi-Know-how für Ihre tägliche Arbeit: in GFA-BASIC Tips & Tricks. Hier verraten Ihnen echte GFA-Spezialisten all ihre kleinen und großen Kniffe, mit denen sie noch mehr aus diesem leistungsstarken BASIC herausholen.
GFA-BASIC 2.0 erforderlich.
GFA-BASIC Tips & Tricks
Hardcover, 350 Seiten
inkl. Diskette, DM 49,–

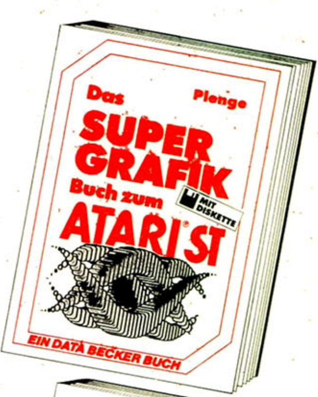


DATA BECKER Führer zum ATARI ST
240 Seiten
DM 29,80



DATA BECKER Führer zu GFA-BASIC
254 Seiten
DM 24,80

Drei Bücher zum ATARI ST ersparen Ihnen das lange Suchen im Handbuch oder in der Fachliteratur. Fehlt Ihnen eine bestimmte Information, können Sie immer ganz gezielt nachschlagen: Die DATA BECKER Führer – die erfolgreichste und kompletteste Serie ihrer Art. Kompetent, zuverlässig und immer griffbereit.



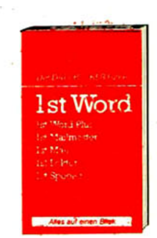
Das Supergrafikbuch zum ST – vollgepackt mit dem Know-how, das jeder engagierte ST-Anwender braucht. Von den Grundlagen bis zu speziellen Problemlösungen wie Programmierung eines Rasterinterrupts oder einer flackerfreien Animation finden Sie hier alles zum Thema Grafik. Mit zahlreichen Utilities in GFA-BASIC, C und Assembler. Ein Buch nicht nur für Grafik-Freaks.
Das Supergrafikbuch zum ATARI ST
Hardcover, 838 Seiten
inkl. Diskette, DM 69,–



Alles zu der Textverarbeitung 1st Word Plus – einschließlich der Zusatzprogramme 1st Mail, 1st Lektor, 1st Proportional und 1st Index. Mit vielen Tips und Lösungen aus der praktischen Arbeit. Nicht nur ein Lehrbuch, sondern auch ein hervorragendes Nachschlagewerk.
Das große Buch zu 1st Word Plus
Hardcover, 288 Seiten
inkl. Diskette, DM 59,–



Intern-Bände von DATA BECKER sind seit jeher Informationspakete ganz besonderer Art. So auch hier: Von der Pinbelegung bis hin zum kommentierten BIOS-Listing wird jedes Detail Ihres Rechners ausführlich beschrieben – natürlich auch der Atari-Blitter. Einfach die Pflichtlektüre für den ST-Profi.
ATARI ST Intern
Hardcover, 637 Seiten
DM 69,–



DATA BECKER Führer zu 1st Word
192 Seiten
DM 24,80

COUPON

HIERMIT BESTELLE ICH

NAME, VORNAME

STRASSE, ORT

zzgl. DM 5,– Versandkosten unabhängig von der bestellten Stückzahl
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

Von indizierten und sequentiellen Zugriffen

Ohne sie beim Namen genannt zu haben, wurden in der heutigen und in der letzten Folge bereits die zwei Arten von Zugriffen auf eine Datenbank behandelt. Es handelt sich dabei um:

A: Indizierte Zugriffe, die schnell über irgendeine Organisationsstruktur (Index) erfolgen (Binärbaum, AVL-Baum, ...).

B: Sequentielle Zugriffe, die die kompletten Daten durcharbeiten und dabei mitunter recht langsam sind. Bei den Baumstrukturen sind dies die X-order-Zugriffe.

Mit einer besonderen sequentiellen Form des Auffindens von Daten möchte ich mich nun befassen.
Ausgangspunkt ist dabei die Idee einer Adreßdatei als Anwendungsbeispiel für Bäume. Da in so einer Adreßdatei sämtliche Daten in Form von Strings gespeichert werden können, werden wir nun eine Möglichkeit behandeln, Daten zu finden, indem wir ein Suchmuster für die entsprechenden Daten vorgeben und während eines X-order-Durchlaufs dieses Suchmuster mit sämtlichen Datensätzen vergleichen. Mächtigkeit erlangt dieses Konzept durch die Möglichkeit Wildcards ('*' und '?') bei den Suchmustern zu verwenden.
Beispiel: Die Typdefinitionen für unsere Adreßdatei, vorausgesetzt, möchten wir uns sämtliche Dortmunder, die Peter heißen und in unserer Adreßdatei gespeichert sind, ausgeben lassen.
Als Suchmuster geben wir also vor:

```
Anrede    : *
Vorname   : *Peter*
Nachname  : *
Straße    : *
Wohnort   : *Dortmund*
Telefon   : *
```

Eine Stringrelation

Was nun benötigt wird, um Datensätze mit derartigen Suchmustern vergleichen zu können, ist im wesentlichen eine Stringrelation, die Wildcards berücksichtigt (Listing 5c, Zeilen 54-107).
Dabei gehen wir davon aus, daß einer der beiden Strings Wildcards enthalten darf, der andere nicht. Beim Vergleich der beiden Strings wird versucht, beide bis auf den leeren String ('') zu reduzieren, indem gleiche Buchstaben abgetrennt werden und mit den Reststrings ein rekursiver Aufruf der Stringrelation erfolgt. Wurde nun in irgendeinem Teil des rekursiven Aufrufbaumes der Prozedur eine Reduktion auf zwei leere Strings erzielt, ist ein Flag zu setzen, das, zum einen den Abbruch der Rekursion veranlaßt, und zum anderen als Gleichheitsbit dient (equal).

```
Beispiel: norm_str = 'abcde'
          wild_str = 'ab?*'
Der rekursive Aufrufbaum
hätte folgendes Aussehen:
     durchlauf('ab?*','abcde')
     durchlauf('b?*','bcde')
     durchlauf('?*','cde')
     durchlauf('*','de')
     durchlauf('','e') und
     durchlauf('','')
```

Hierbei führt der zweite Teil des durch '*' verursachten Doppelaufrufs zu einer Auslöschung der beiden Strings, also zur Vergleichbarkeit der ursprünglichen Strings.

muster_inorder

Zum sequentiellen Durchlauf der Baumstruktur habe ich den inorder-Durchlauf (Zeilen 109-132) gewählt. Zusätzlich zu den beiden Teilbäumen ist dabei noch die Nachfolgerliste zu durchlaufen (ebenfalls rekursiv).
Auf den Knoten wird in inorder-Position die Stringrelation auf sämtlichen Schlüssel- und Datenstrings ausgeführt. Dabei werden die entsprechenden Knoteninhalte mit den Werten einer globalen Variable muster verglichen. Bei der Variable muster handelt es sich um einen b_file_type, da die Organisationsvariablen für die Baumstruktur weder bei den Dateioperationen noch bei den Suchmustern benötigt werden.

Als Operation, die in die Rekursion eingeschachtelt ist, habe ich eine Elementausgabe gewählt. Denkbar sind natürlich alle möglichen Operationen, denen lediglich gemeinsam ist, daß sie die Baumstruktur festhalten, also beispielsweise 'Editieren des Datensatzes'. Dabei ist allerdings darauf zu achten, daß der Schlüssel nicht verändert wird, denn sonst geht die Ordnung der Baumstruktur verloren.

eingabe und ausgabe

Zwei Operationen, die ebenfalls nicht fehlen dürfen, sind die beiden Operationen eingabe und ausgabe zum Lesen und Schreiben eines Datensatzes. (eingabe wird auch zur Mustereingabe verwendet.)

main

Das Hauptprogramm beinhaltet, organisiert in einem zentralen CASE-Statement, die sieben Operationen,
Eingabe, Löschen, Suche nach Muster, Suche nach Nachname, Speichern, Laden und **Löschen.**
Einen etwas mehr als trivialen Operationsaufruf bieten aber nur die beiden Operationen Löschen und Suche nach Nachname.
Bei Löschen (CASE-Label 2) wird nach dem Ausfügen aus dem Baum die dabei ausgefügte Elementenliste durchlaufen. Die Datensätze, die dabei als wertvoll erachtet werden, werden erneut in die Baumstruktur eingefügt, die anderen werden freigegeben.
Bei der Suche nach Nachnamen (CASE-Label 4) ist nach dem Auffinden eines Elementes mit search_first die Elementnachfolgerliste mit search_next zu durchlaufen.

Vorausschau

In der nächsten Folge werde ich Sie mit einem etwas anderen Ansatz zur Suchpfadoptimierung in Bäumen bekanntmachen, mit den sogenannten B-Bäumen. Denjenigen unter Ihnen, denen die Botanik langsam zum Hals heraushängt, sei versichert, daß es sich dabei um den letzten Vertreter der Spezies Baum handeln wird.

Dirk Brockhaus

Kieckbusch liefert

TIMEWORKS DTP

Das deutsche TIMEWORKS DTP besitzt alle Funktionen, die Sie benötigen, um professionelle Vorlagen zu fertigen. Es ist einfach zu bedienen und kann auf dem Atari ST oder IBM PC und kompatiblen Computern verwendet werden. Es ist sehr vielseitig und mit vielen der schon existierenden populären Sofware-Programme kompatibel.

☞ **Kieckbusch Aktion**

Senden Sie uns Ihr Original -DTP- Programm (z.B. Becker Page, oder Publishing-Partner usw.) und 189,- DM und Sie erhalten das TIMEWORKS DTP von uns.

Preis
DM 389,-

* Diese Anzeige wurde mit dem Timeworks Publisher erstellt

Steve 3.0 kann lesen !

Text • Grafik • Datenbank

- Das neue Steve 3.0 beinhaltet jetzt auch Schrifterkennung
- speziell mit Hawk-Scanner und Handy-Scanner zu empfehlen

Preis inkl.
Texterkennung
DM 498,-
Update DM 100,-

Handy Scanner Type II

Scan-Breite:	64mm,512 Punkte/Zeile
Auflösung:	horizontal 8 Punkte/mm
Kontrast:	vertikal 8 Punkte/mm
Software:	Empfindlichkeit manuell einstellbar

zur Schrifterkennung mit Steve

Weitere professionelle Kieckbusch-Produkte

- ☆ VIP Professional
- ☆ Desk Assist II***
- ☆ LOGISTIX
- ☆ Multi-Hardcopy
- ☆ WordPerfect

Computer Technik Kieckbusch GmbH
Baumstammhaus 5419 Vielbach
☎ 02626-78336 & 8991 FAX:78337

Fragen Sie Ihren Händler !
Österreich: Überreuter Media, Wien
Schweiz: Senn-Computer, Zürich

AVL_TYP.PAS

```
{ Listing 5a)

   Typen zu den AVL-Baeumen

    Reservierte Woerter : b_key_type, b_data_type,
                          b_tree, b_element
                          und b_file_type

   programmiert : Juni 87
   von          : D. Brockhaus
   mit          : PASCAL+ (CCD)
}

b_key_type   = string;

b_data_type  = RECORD
                 anrede  ,
                 vorname ,
                 strasse ,
                 wohnort ,
                 telefon : string;
               END;

b_tree       = ^b_element;

b_element    = RECORD
                 key    : b_key_type;
                 data   : b_data_type;
                 bal    : -1..+1;
                 left   ,
                 right  ,
                 next   : b_tree;
               END;

b_file_type  = RECORD
                 key  : b_key_type;
                 data : b_data_type;
               END;
```

AVL_TREE.PAS

```
{ Listing 5b)

   Operationen auf AVL-Baeumen

   Funktionalitaeten :

   ( - io_check      : boolean -> () )
   ( - io_result     : () -> integer )
     - create        : () -> b_tree
     - insert        : b_tree x b_key_type x b_data_type
                          -> b_tree x boolean
     - remove        : b_tree x b_key_type
                          -> b_tree x b_tree
   ( - search_ptr    : b_tree x b_key_type
                          -> b_tree x boolean )
     - search_first  : b_tree x b_key_type
                          -> b_data_type x boolean
     - search_next   : b_tree x b_key_type
                          -> b_data_type x boolean
     - save          : b_tree x string -> file x boolean
     - load          : file x string -> b_tree x boolean

   Reservierte Woerter : present_ptr, io_check,
                         io_result, create,
                         insert, remove, search_ptr,
                         search_first,
                         search_next, save und load.

  programmiert : Juni 87
  von          : D. Brockhaus
  mit          : PASCAL+ (CCD)
}

present_ptr : b_tree;

PROCEDURE io_check(checked : boolean);
  EXTERNAL;

FUNCTION io_result : integer;
  EXTERNAL;

PROCEDURE create(VAR tree : b_tree);

  BEGIN {create}
    tree:=nil;
  END;  {create}

FUNCTION insert(VAR tree      : b_tree;
                    new_key   : b_key_type;
                    new_data  : b_data_type) : boolean;

  VAR element  : b_tree;
      store    ,
      help     : boolean;

  PROCEDURE put_in(     element  : b_tree;
                    VAR p        : b_tree;
                    VAR h        : boolean);

    VAR p1 ,
        p2 : b_tree;

    BEGIN {put_in}
      IF p=nil THEN
        BEGIN
          h:=true;
          p:=element;
        END
      ELSE
        IF element^.key<p^.key THEN
          BEGIN
            put_in(element,p^.left,h);
            IF h THEN
              CASE p^.bal OF
                1 : BEGIN
                      p^.bal:=0;
                      h:=false;
                    END;
                0 : p^.bal:=-1;
               -1 : BEGIN
                      p1:=p^.left;
                      IF p1^.bal=-1 THEN
                        BEGIN
                          p^.left:=p1^.right;
                          p1^.right:=p;
                          p^.bal:=0;
                          p:=p1;
                        END
                      ELSE
                        BEGIN
                          p2:=p1^.right;
                          p1^.right:=p2^.left;
                          p2^.left:=p1;
                          p^.left:=p2^.right;
                          p2^.right:=p;
                          IF p2^.bal=-1 THEN
                            p^.bal:=1
                          ELSE
                            p^.bal:=0;
                          IF p2^.bal=1 THEN
                            p1^.bal:=-1
                          ELSE
                            p1^.bal:=0;
                          p:=p2;
                        END;
                      p^.bal:=0;
                      h:=false;
                    END;
              END;
          END
        ELSE
          IF element^.key>p^.key THEN
            BEGIN
              put_in(element,p^.right,h);
              IF h THEN
                CASE p^.bal OF
                 -1 : BEGIN
                        p^.bal:=0;
                        h:=false;
                      END;
                  0 : p^.bal:=1;
                  1 : BEGIN
                        p1:=p^.right;
                        IF p1^.bal=1 THEN
                          BEGIN
                            p^.right:=p1^.left;
                            p1^.left:=p;
                            p^.bal:=0;
                            p:=p1;
```

geht weiter...

```
                                END
                              ELSE
                                BEGIN
                                   p2:=p1^.left;
                                   p1^.left:=p2^.right;
                                   p2^.right:=p1;
                                   p^.right:=p2^.left;
                                   p2^.left:=p;
                                   IF p2^.bal=1 THEN
                                     p^.bal:=-1
                                  ELSE
                                     p^.bal:=0;
                                   IF p2^.bal=-1 THEN
                                     p1^.bal:=1
                                  ELSE
                                     p1^.bal:=0;
                                   p:=p2;
                                END;
                              p^.bal:=0;
                              h:=false;
                            END;
                    END;
               END
             ELSE
                BEGIN
                  h:=false;
                   element^.next:=p^.next;
                   p^.next:=element;
                END;
     END;  {put_in}

   BEGIN {insert}
       store:=memavail>=2*sizeof(b_element);
     IF store THEN
       BEGIN
          new(element);
          WITH element^ DO
            BEGIN
               key:=new_key;
               data:=new_data;
               bal:=0;
               right:=nil;
               left:=nil;
               next:=nil;
            END;
          help:=false;
          put_in(element,tree,help);
       END;
     insert:=store
   END;  {insert}

PROCEDURE remove(VAR tree       : b_tree;
                     key        : b_key_type;
                 VAR del_list   : b_tree);

   VAR help : boolean;

   PROCEDURE delete(    key : b_key_type;
                    VAR p   : b_tree;
                    VAR h   : boolean);

     VAR q : b_tree;

      PROCEDURE balance1(VAR p : b_tree;
                         VAR h : boolean);

       VAR p1 ,
            p2 : b_tree;
            b1 ,
            b2 : -1..+1;

        BEGIN {balance1}
          CASE p^.bal OF
           -1 : p^.bal:=0;
            0 : BEGIN
                   p^.bal:=1;
                   h:=false;
                 END;
            1 : BEGIN
                   p1:=p^.right;
                   b1:=p1^.bal;
                   IF b1>=0 THEN
                     BEGIN
                        p^.right:=p1^.left;
                        p1^.left:=p;
                         IF b1=0 THEN
                           BEGIN
                             p^.bal:=1;
                             p1^.bal:=-1;
                             h:=false;
                           END
                         ELSE
                           BEGIN
                             p^.bal:=0;
                             p1^.bal:=0;
                           END;
                         p:=p1;
                      END
                    ELSE
                      BEGIN
                         p2:=p1^.left;
                         b2:=p2^.bal;
                         p1^.left:=p2^.right;
                         p2^.right:=p1;
                         p^.right:=p2^.left;
                         p2^.left:=p;
                         IF b2=1 THEN
                           p^.bal:=-1
                         ELSE
                           p^.bal:=0;
                         IF b2=-1 THEN
                           p1^.bal:=1
                         ELSE
                           p1^.bal:=0;
                         p:=p2;
                         p2^.bal:=0;
                      END;
                 END;
          END;
        END;  {balance1}

      PROCEDURE balance2(VAR p : b_tree;
                         VAR h : boolean);

        VAR p1 ,
             p2 : b_tree;
             b1 ,
             b2 : -1..+1;

        BEGIN {balance2}
          CASE p^.bal OF
             1 : p^.bal:=0;
             0 : BEGIN
                    p^.bal:=-1;
                    h:=false;
                 END;
            -1 : BEGIN
                    p1:=p^.left;
                    b1:=p1^.bal;
                    IF b1<=0 THEN
                      BEGIN
                          p^.left:=p1^.right;
                         p1^.right:=p;
                         IF b1=0 THEN
                           BEGIN
                              p^.bal:=-1;
                              p1^.bal:=1;
                              h:=false;
                           END
                         ELSE
                           BEGIN
                              p^.bal:=0;
                              p1^.bal:=0;
                           END;
                         p:=p1;
                      END
                    ELSE
                      BEGIN
                         p2:=p1^.right;
                         b2:=p2^.bal;
                         p1^.right:=p2^.left;
                         p2^.left:=p1;
                         p^.left:=p2^.right;
                         p2^.right:=p;
                         IF b2=-1 THEN
                           p^.bal:=1
                         ELSE
                           p^.bal:=0;
                         IF b2=1 THEN
                           p^.bal:=-1
                         ELSE
```

geht weiter...

```
                    p^.bal:=0;
                  p:=p2;
                  p2^.bal:=0;
                END;
              END;
          END;
        END;  {balance2}

      PROCEDURE del(VAR r : b_tree;
                    VAR h : boolean);

        BEGIN {del}
          IF r^.right<>nil THEN
            BEGIN
              del(r^.right,h);
              IF h THEN
                balance2(r,h);
            END
          ELSE
            BEGIN
              q^.key:=r^.key;
              q^.data:=r^.data;
              q^.next:=r^.next;
              q:=r;
              r:=r^.left;
              h:=true;
            END;
        END;  {del}

      BEGIN {delete}
        IF p=nil THEN
          h:=false
        ELSE
          IF key<p^.key THEN
            BEGIN
              delete(key,p^.left,h);
              IF h THEN
                balance1(p,h);
            END
          ELSE
            IF key>p^.key THEN
              BEGIN
                delete(key,p^.right,h);
                IF h THEN
                  balance2(p,h);
              END
            ELSE
              BEGIN
                q:=p;
                IF q^.right=nil THEN
                  BEGIN
                    p:=q^.left;
                    h:=true;
                  END
                ELSE
                  IF q^.left=nil THEN
                    BEGIN
                      p:=q^.right;
                      h:=true;
                    END
                  ELSE
                    BEGIN
                      del(q^.left,h);
                      IF h THEN
                        balance1(p,h)
                    END;
                del_list:=q;
              END;
      END;  {delete}

    BEGIN {remove}
      help:=false;
      del_list:=nil;
      delete(key,tree,help);
    END;  {remove}

  FUNCTION search_ptr(     tree : b_tree;
                           key  : b_key_type;
                       VAR list : b_tree) : boolean;

    BEGIN {search_ptr}
      IF tree<>nil THEN
        IF key<tree^.key THEN
          search_ptr:=search_ptr(tree^.left,key,list)
        ELSE
          IF key>tree^.key THEN
            search_ptr:=search_ptr(tree^.right,key,list)
          ELSE
            BEGIN
              search_ptr:=true;
              list:=tree;
            END
      ELSE
        search_ptr:=false;
    END;  {search_ptr}

  FUNCTION search_first(     tree : b_tree;
                             key  : b_key_type;
                         VAR data : b_data_type) :
                                    boolean;

    VAR found : boolean;

    BEGIN {search_first}
      found:=search_ptr(tree,key,present_ptr);
      IF found THEN
        data:=present_ptr^.data;
      search_first:=found;
    END;  {search_first}

  FUNCTION search_next(VAR data : b_data_type) : boolean

    VAR found : boolean;

    BEGIN {search_next}
      present_ptr:=present_ptr^.next;
      found:=present_ptr<>nil;
      IF found THEN
        data:=present_ptr^.data;
      search_next:=found;
    END;  {search_next}

  FUNCTION save(tree       : b_tree;
                filename : string) : boolean;

    VAR t  : FILE OF b_file_type;
        ok : boolean;

    PROCEDURE put_praeorder(tree : b_tree);

      VAR help : b_tree;

      BEGIN {put_praeorder}
        IF tree<>nil THEN
          BEGIN
            help:=tree;
            WHILE (help<>nil) AND ok DO
              BEGIN
                t^.key:=tree^.key;
                t^.data:=tree^.data;
                help:=help^.next;
                put(t);
                ok:=io_result=0;
              END;
            IF ok THEN
              BEGIN
                put_praeorder(tree^.left);
                put_praeorder(tree^.right);
              END;
          END;
      END;  {put_praeorder}

    BEGIN {save}
      io_check(false);
      rewrite(t,filename);
      ok:=io_result=0;
      IF ok THEN
        put_praeorder(tree);
      save:=ok;
      io_check(true);
    END;  {save}

  FUNCTION load(VAR tree       : b_tree;
                    filename : string) : boolean;

    VAR t  : FILE OF b_file_type;
        ok : boolean;

    BEGIN {load}
      io_check(false);
      create(tree);
      reset(t,filename);
      ok:=io_result=0;
      IF ok THEN
```

gehat weiter...

```
        WHILE NOT eof(t) AND ok DO
          BEGIN
            ok:=insert(tree,t^.key,t^.data);
            IF ok THEN
              BEGIN
                get(t);
                ok:=io_result=0;
              END;
          END;
      load:=ok;
      io_check(true);
    END;  {load}
```

AVL_TEST.PAS

```
{ Listing 5c)

   Eine Adressdatei mit AVL-Trees.

   programmiert : Oktober 87
   von          : D. Brockhaus
   mit          : PASCAL+ (CCD)
}

 PROGRAM test_avl_trees (input,output);

 TYPE {$i avl_typ.pas}

 VAR  tree      ,
      list      ,
      old       : b_tree;
      muster    : b_file_type;
      key       : b_key_type;
      data      : b_data_type;
      str       : string;
      question : char;

      {$i avl_tree.pas}

 PROCEDURE eingabe(VAR key  : b_key_type;
                   VAR data : b_data_type);

   BEGIN {eingabe}
     WITH data DO
       BEGIN
         write(' Anrede....  '); readln(anrede);
         write(' Vorname...  '); readln(vorname);
         write(' Nachname..  '); readln(key);
         write(' Strasse...  '); readln(strasse);
         write(' Wohnort...  '); readln(wohnort);
         write(' Telefon...  '); readln(telefon);
       END;
   END;  {eingabe}

 PROCEDURE ausgabe(key   : b_key_type;
                   data  : b_data_type);

   BEGIN {ausgabe}
     WITH data DO
       BEGIN
         writeln(' Anrede....  ',anrede);
         writeln(' Vorname...  ',vorname);
         writeln(' Nachname..  ',key);
         writeln(' Strasse...  ',strasse);
         writeln(' Wohnort...  ',wohnort);
         writeln(' Telefon...  ',telefon);
       END;
   END;  {ausgabe}

 FUNCTION str_rel(wild_str ,
                  norm_str : string) : boolean;

   VAR equal : boolean;

   PROCEDURE durchlauf(wild ,
                       norm : string);

     BEGIN {durchlauf}
       IF NOT equal THEN
         IF (wild='') THEN
           equal:=norm=''
         ELSE
           IF norm='' THEN
             BEGIN
               IF wild[1]='*' THEN
                 BEGIN
                   delete(wild,1,1);
                   durchlauf(wild,norm);
                 END;
             END
           ELSE
             CASE wild[1] OF
               '*' : BEGIN
                       REPEAT
                         IF wild<>'' THEN
                           delete(wild,1,1);
                       UNTIL (wild[1]<>'*') OR
                             (wild='');
                       REPEAT
                         durchlauf(wild,norm);
                         delete(norm,1,1);
                       UNTIL norm='';
                       durchlauf(wild,norm);
                     END;
               '?' : BEGIN
                       delete(wild,1,1);
                       delete(norm,1,1);
                       durchlauf(wild,norm);
                     END;
               OTHERWISE :
                     IF norm[1]=wild[1] THEN
                       BEGIN
                         delete(wild,1,1);
                         delete(norm,1,1);
                         durchlauf(wild,norm);
                       END;
             END;
     END;  {durchlauf}

   BEGIN {string_relation}
     equal:=false;
     durchlauf(wild_str,norm_str);
     str_rel:=equal;
   END;  {string_relation}

 PROCEDURE muster_inorder(VAR tree  : b_tree);

   BEGIN {muster_inorder}
     IF tree<>nil THEN
       BEGIN
         IF tree^.left<>nil THEN
           muster_inorder(tree^.left);
         IF str_rel(muster.key,tree^.key) AND
            str_rel(muster.data.anrede,
               tree^.data.anrede) AND
            str_rel(muster.data.vorname,
               tree^.data.vorname) AND
            str_rel(muster.data.strasse,
               tree^.data.strasse) AND
            str_rel(muster.data.wohnort,
               tree^.data.wohnort) AND
            str_rel(muster.data.telefon,
               tree^.data.telefon) THEN
           BEGIN
             ausgabe(tree^.key,tree^.data);
             writeln('Weiter mit <RETURN>');
             readln;
           END;
         IF tree^.next<>nil THEN
           muster_inorder(tree^.next);
         IF tree^.right<>nil THEN
           muster_inorder(tree^.right);
       END;
   END;  {muster_inorder}

 BEGIN {main}
   create(tree);
   REPEAT
     writeln;
     writeln('Hauptmenue');
     writeln;
     writeln('<1> Einfuegen');
     writeln('<2> Loeschen');
     writeln('<3> Suche nach Muster');
     writeln('<4> Suche nach Nachname');
     writeln('<5> Datei speichern');
     writeln('<6> Datei laden');
     writeln('<7> Datei initialisieren');
     writeln('<0> Beenden');
     write(' > ');
     read(question);
     writeln;
```

geht weiter...

```
      CASE question OF
        '1' : BEGIN
                writeln;
                writeln('Eingabe');
                writeln;
                eingabe(key,data);
                IF NOT insert(tree,key,data) THEN
                  writeln('Speicher ist voll !!!');
              END;
        '2' : BEGIN
                writeln;
                writeln('Loeschen');
                writeln;
                write(' Nachname.. '); readln(key);
                remove(tree,key,list);
                WHILE list<>nil DO
                  BEGIN
                    old:=list;
                    list:=list^.next;
                    ausgabe(old^.key,old^.data);
                    write('Wollen Sie diesen Datensatz
                           loeschen (j,n) ?');
                    readln(question);
                    IF question IN ['n','N'] THEN
                      IF insert(tree,old^.key,
                         old^.data) THEN
                        ;
                    dispose(old);
                  END;
              END;
        '3' : BEGIN
                writeln;
                writeln('Suche nach Muster');
                writeln;
                writeln('Eingabe Muster');
                writeln;
                eingabe(muster.key,muster.data);
                muster_inorder(tree);
              END;
        '4' : BEGIN
                writeln;
                writeln('Suche nach Nachname');
                writeln;
                write(' Nachname.. '); readln(key);
                writeln;
                IF search_first(tree,key,data) THEN
                  BEGIN
                    ausgabe(key,data);
                    writeln('Weiter mit <RETURN>');
                    readln;
                    WHILE search_next(data) DO
                      BEGIN
                        ausgabe(key,data);
                        writeln('Weiter mit <RETURN>');
                        readln;
                      END;
                    writeln('Ende der Liste');
                    readln;
                  END;
              END;
        '5' : BEGIN
                writeln;
                writeln('Speichern');
                writeln;
                write(' Dateiname.. '); readln(str);
                IF NOT save(tree,str) THEN
                  writeln('Fehler beim Speichern');
              END;
        '6' : BEGIN
                writeln;
                writeln('Laden');
                writeln;
                write(' Dateiname.. '); readln(str);
                IF NOT load(tree,str) THEN
                  writeln('Fehler beim Laden');
              END;
        '7' : create(tree);
      END;
    UNTIL question='0';
  END.  {main}
```

Wie Sie sicherlich bemerkt haben, ist das letzte Heft wieder pünktlich erschienen. Dabei ist leider dieses Listing aus der letzten Folge der Algorithmen & Datenstrukturen unter die Räder geraten. Wir bitten dafür um Entschuldigung.

Fehlendes Listing aus Teil 4

```
{ Listing 4c)

   Programm test_b_trees liefert eine
   Testumgebung zu den b_tree Modulen.
   Es wird ein Baum verwaltet, in dem

     - eingefuegt und geloescht werden kann
     - eine Suche nach Daten gleichen
       Schluessels durchgefuehrt werden kann
     - Dateioperationen moeglich sind

   Weiterhin koennen, zusaetzlich zu den
   Modulfunktionen prae-, post-, und
   inorder Durchlaeufe gemacht werden.

   programmiert  : Juni 87
   von           : Dirk Brockhaus
   mit           : PASCAL+ (CCD)
}

PROGRAM test_b_trees (input,output);

TYPE {$i baum_typ.pas}

VAR tree      : b_tree;
    key       : key_type;
    data      : data_type;
    str       : string;
    question  : char;

{$i baum.pas}

PROCEDURE drucke_daten(liste : b_tree);

  BEGIN {drucke_daten}
    WHILE liste<>nil DO
      BEGIN
        write(' ',liste^.data);
        liste:=liste^.next;
      END;
  END;  {drucke_daten}

PROCEDURE inorder(tree : b_tree);

  BEGIN {inorder}
    IF tree<>nil THEN
      BEGIN
        inorder(tree^.left);
        write(' Schluessel: ',tree^.key,
              ' Daten: ');
        drucke_daten(tree);
        writeln;
        inorder(tree^.right);
      END;
  END;  {inorder}

PROCEDURE praeorder(tree : b_tree);

  BEGIN {praeorder}
    IF tree<>nil THEN
      BEGIN
        write(' Schluessel: ',tree^.key,
              ' Daten: ');
        drucke_daten(tree);
        writeln;
        praeorder(tree^.left);
        praeorder(tree^.right);
      END;
  END;  {praeorder}

PROCEDURE postorder(tree : b_tree);

  BEGIN {postorder}
    IF tree<>nil THEN
      BEGIN
        postorder(tree^.left);
        postorder(tree^.right);
        write(' Schluessel: ',tree^.key,
              ' Daten: ');
```

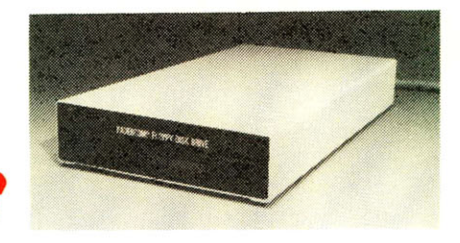
geht weiter...

PADERCOMP — Walter Ladz

Erzbergerstraße 27 · 4790 Paderborn · Tel. 0 52 51-3 63 96

Sie sind da, die neuen PADERCOMPS

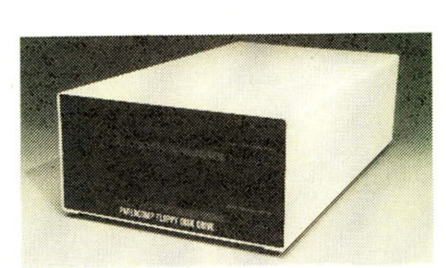
- noch kleiner
- noch leiser
- noch zuverlässiger
- noch günstiger
- mit NEC FD 1037 A



PADERCOMP FL-1

3,5", 1 MB, NEC FD 1037 A,
anschlußfertig für ATARI ST
Abm.: 153 × 103 × 26 mm
incl. Steckernetzteil PPS-1

348.-*



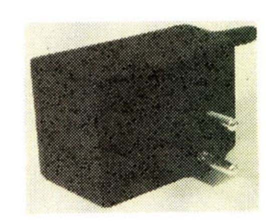
PADERCOMP FL-2

Doppelstation übereinander,
Abm.: 153 × 103 × 52 mm
incl. Steckernetzteil PPS-1

598.-*

Sie erhalten diese Qualitätsprodukte nur bei speziellen Fachhändlern oder direkt von uns. Rufen Sie an!

Händleranfragen erwünscht



Steckernetzteil PPS-1

5 V/1 A stabilisiert, kurzschlußfest, eigens für die neuen NEC FD 1037 entwickelt, serienmäßig bei unseren Laufwerken FL-1 und FL-2 enthalten. Einzelpreis **49.-***

* unverbindliche Preisempfehlung

Diskstation LA1 1 MB, anschluß. 298,-

Neue Generation mit 3,5" **NEC 1037-Laufw.**, im grauen Alugehäuse ★ inclusive Netzteil ★ 100% kompatibel ★ Extrem leise und klein ★ Maße in cm: L 17,5 x B 10,4 x H 2,9 ★ Für alle ST u. Mega ST ★ **ACHTUNG: Unsere Stationen laufen auch als 2. Station an SF 314/354 und 1040 ST einwandfrei**

Gegen Aufpreis in mehreren Ausführungen lieferbar:
★ Ein-/Ausschalter DM 10,– ★ Netzkontroll-Leuchte DM 10,–
★ Buchse zum Anschluß einer weiteren Station DM 35,– ★

Doppelstation 2xLA1 598,-

Floppybox 3 LAUFWERKE AM ST 99,-

Ermöglicht den Anschluß von 3 Stationen am ST, integrierte **Treiberschaltung** für 5,25" Laufwerke ★ **Leuchtdioden** zeigen an, welche Laufwerke aktiviert sind ★ Umschaltung während des Betriebs ohne Reset und Programmverlust ★ SF314/354-Laufwerke auch als B- oder C-Station anschließbar ★ Für alle ST/Mega ST ★ Bei Bestellung **Rechnertyp angeben!** ★

21 MB Festplatte 1.198,-

Vortex HD plus 20 im neuen Mega ST Design ★ Bootfähig ★ Anschlußfertig und partioniert ★ inclusive Disk-Backup-Programm, Festplattenmanager, Auto-Park-Programm, Catch-Memory ★ Für alle ST und Mega ST ★

Monitorbox anschlußfertig 49,90

Umschaltung zwischen Farb- u. S/W-Monitor (Bild u. Ton) **anschlußfertig im Gehäuse eingebaut** ★ Mit ca. 20 cm langem Kabel, d. h. Rechnerbuchsen werden nicht verdeckt.

ST-Oszilloskop 399,-

ST als Oszilloskop, Speicheroszilloskop, und Oszillograph. 50 Seiten Bildschirmspeicher ★ Meßdauer: 1 msec bis 70 Std. 60 000 Messungen pro Sekunde ★ Timebase: 50us/L bis 500 sec/L ★ Die gespeicherten Daten können mit beliebiger Software weiterverarbeitet werden ★ Auch als Sound-Sampler einsetzbar ★ Mit zusätzlicher Software ist der Betrieb als EKG-Auswerter, Sprachanalysator, Nachhallmeßgerät usw. möglich.

Anschlußfertige Kabel

Festplatten-Verlängerungskabel, 1 m, 49,90 ★ **Festplatten-Ersatzkabel**, 2 m, 55,– ★ **Floppykabel (3,5")** 29,90 ★ **Floppykabel (5,25")** 29,90 ★ dto. für 2 Laufwerke 49,90 ★ **Monitorkabel** für Fremdmonitor (Cinch) Rechnertyp angeben 35,– ★ **Scartkabel, 2 m** (V), mit R-Schaltung 39,90 ★ **Druckerkabel 2 m, rund 29,90** ★ **Midi-Kabel, 5 m, 35,–** ★

Stecker und Buchsen

Original 40pol **Modulportbuchse** 19,– ★ **Monitorstecker** 6,90 ★ **Monitorbuchse** 8,90 ★ **Floppybuchse** 10,90 ★ **Floppystecker** 8,90 ★ 19pol **DMA-Stecker** mit Gehäuse 9,90 ★ Sonstige auf Anfrage

Sonstiges Zubehör

Speichererweiterung auf 1 MB, 2 MB u. 4 MB auf Anfrage ★ **Uhrmodul** 99,– ★ Floppy-**Einbaunetzteil** für 3,5" u. 5,25" Laufwerke (12 V, 5 V/5 W) 45,– ★ **Alu Rohgehäuse für NEC 1037** 19,– ★ **Blittertos-Fastrom** 25,– ★ **Fastrom** (altes TOS) 25,– ★ Betriebssystem incl. Fastrom 99,– ★ Blitter-Betriebssystem incl. Fastrom 99,– ★ Blitter-IC (Nachrüsten aller älteren ST) Preis auf Anfrage ★ **NEC 1037** (incl. Anschlußkabel u. Powerstecker 249,– ★ Demnächst lieferbar: Handy-Scanner und MS DOS Emulator (Hard- und Software-Vers.)

Versandhinweis: Mindestbestellwert DM 20,– ★ Versandkosten: Nachnahme DM 6,50 ★ Vorkasse DM 5,– ★ Versand ins Ausland nur gegen Vorkasse und abzüglich 14 % Mwst. (Versandkosten DM 12,–)

WISCHOLEK Computertechnik ★ **Mesteroth 9 ★ 4250 Bottrop 2 ★ ☎ (0 20 45) 8 16 38** Mo–Fr 10–11.30 u. 15–18 Uhr
VORMALS MIWIKO ★ NUR VERSAND: DIREKTVERKAUF UND VORFÜHRUNGEN NUR NACH VORHERIGER TELEFONISCHER TERMINVEREINBARUNG.



ruff & Locher datentechnik NEU!

Eichachstraße 13, 7404 Ofterdingen, Tel. (07473) 22810

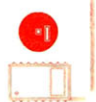
endlich! »ST-TAST« endlich!

ermöglicht es, Tastaturen im IBM-Design am ATARI zu betreiben.

- Die Tastatur wird einfach über unser ST-TAST Interface am Modulport eingesteckt.
- Die Treibersoftware ist autostartfähig.
- Auch eine extravagante Tastenbelegung ist über die mitgelieferte Software problemlos zu realisieren.

Wir bieten an:

- IBM-Kompatible Tastaturen, Preise auf Anfrage
- ST-TAST Interface incl. Software und ausführliche Anleitung **DM 94,-**
- fordern Sie unser Informationsmaterial an.



POWER WITHOUT THE PRICE

GARANTIERT TAGES TIEFST PREISE AUF TEL. ANFRAGE - HÄNDLERANFRAGEN WILLKOMMEN

NEC FD 1037A - Die neue Generation der Disk-Drives: **superflach** (2,54 cm und **100% ST-kompatibel**. Lieferbar als **betriebsfertige** Einzel- oder Doppelstation (auch **Bausätze**) zu traumhaften Preisen.

NEC MULTISYNC GS - Nur noch **ein** Monitor für **alle** ST-Auflösungen (Test in ST 1/88) Lieferung **komplett anschlußfertig** (incl. Tonteil), dt. Modell.

EPSON LQ-500 - Die **brandneue** Alternative unter den 24-Nadlern, **2 NLQ-Modi**, Originalmodell incl. **dt. Handbuch** und **Anschlußkabel** für ST

HARD-DISK 32 MB - Komplettes HD-System mit **vielen Extras** (z.B. Backup-Software kostenlos, Einbaumögl. für 5.25" oder 3.5"-Laufwerke, u.v.m.) zum konkurrenzlosen Preis. **Ausführliche Infos auf Anfrage!**

Nahezu das gesamte ST- und Zubehörangebot zu **günstigsten Preisen** lieferbar. **Rufen Sie unverbindlich an!**

DATENSYSTEME STEFAN KOPPING - STEINWEG 11 - 6312 LAUBACH ☎ / BTX 06405-3350

```
          drucke_daten(tree);
         writeln;
       END;
   END;  {postorder}

BEGIN {main}
  writeln('Test der B-Bäume');
  writeln;
  create(tree);
  REPEAT
    writeln('<1> Einfügen');
    writeln('<2> Löschen eines Schlüssels');
    writeln('<3> Suchen nach
            Schlüsselkriterium');
    writeln('<4> Inorder Baumdurchlauf');
    writeln('<5> Präorder Baumdurchlauf');
    writeln('<6> Postorder Baumdurchlauf');
    writeln('<7> Baum speichern');
    writeln('<8> Baum laden');
    writeln('<9> Baum initialisieren');
    writeln('<0> Beenden');
    write(' > ');
    read(question);
    writeln;
    CASE question OF
      '1' : BEGIN
              write(' Schlüssel > ');
              readln(key);
              write(' Daten > ');
              readln(data);
              insert(tree,key,data);
            END;
      '2' : BEGIN
              write(' Schlüssel > ');
              readln(key);
              IF NOT remove(tree,key) THEN
                BEGIN
                  writeln(' Dieser Schlüssel
                       existierte nicht.');
                  readln;
                END;
            END;
      '3' : BEGIN
              write(' Schlüssel > ');
              readln(key);
              IF search_first(tree,key,data)
             THEN
               BEGIN
                 writeln(' Schlüsselliste : ',
                       key);
                 writeln;
                 writeln(' Daten: ');
                 writeln(' ',data);
                 WHILE search_next(data) DO
                   writeln(' ',data);
               END
             ELSE
               writeln(' Schlüssel nicht
                     gefunden!');
            END;
      '4' : BEGIN
              writeln(' inorder-Durchlauf:');
              inorder(tree);
              readln;
            END;
      '5' : BEGIN
              writeln(' präorder-Durchlauf:');
              praeorder(tree);
              readln;
            END;
      '6' : BEGIN
              writeln(' postorder-Durchlauf:');
              postorder(tree);
              readln;
            END;
      '7' : BEGIN
              write(' Dateiname > ');
              readln(str);
              save(tree,str);
            END;
      '8' : BEGIN
              write(' Dateiname > ');
              readln(str);
              load(tree,str);
            END;
      '9' : create(tree);
    END;
  UNTIL question='0';
END.  {main}
```

ENDE

JURISTISCHE ANWENDUNGS-PROGRAMME

Teil I.

Der gegenwärtige Stand der Unterstützung juristischer Arbeit durch Computer in den verschiedenen Tätigkeitsfeldern

Der Erfolg des Atari ST Computers hängt entscheidend davon ab, ob er sich neben Heimanwendungen auch im wissenschaftlichen und geschäftlichen Bereich durchsetzen kann. Während im wissenschaftlichen Bereich eher die Leistungsfähigkeit der Hardware im Vordergrund steht - wo sich der Atari ST nicht zu verstecken braucht - hängt der Einsatz im geschäftlichen Bereich überwiegend von der zur Verfügung stehenden Software ab.
In letzter Zeit sind mehrere juristische Anwendungsprogramme auf den Markt gekommen, die überwiegend den Anspruch erheben, den Juristen bei seiner Arbeit zu unterstützen: Die Fundstellendateien BGH-DAT und WEHRDAT, Programme für Rechtsanwälte zur Erstellung von Mahnbescheidsanträgen und zur Kostenabrechnung, sowie Textmuster in den Bereichen Unfall-, Familien-, Miet- und Prozeßrecht. Vorwiegend an Jurastudenten und Rechtsreferendare wendet sich dagegen die JURA-Lern- und Definitionskartei als Computerversion für den Atari ST. Es ist daher Zeit für einen ersten Vergleich anhand der Frage, ob diese Programme professionellen Ansprüchen standhalten. Zu diesem Zweck wird in diesem Heft zunächst eine Einführung in die Möglichkeiten der Unterstützung juristischer Arbeit durch den Computer gegeben. Im Anschluß daran sollen in den nächsten Heften die verschiedenen Programme einzeln vorgestellt werden.
Genausowenig, wie es eine einheitliche juristische Arbeitsweise gibt, gibt es einen einheitlichen Standard der Unterstützung dieser Arbeit durch Computer. Es erscheint mir daher sinnvoll, zumindest grob zwischen den Tätigkeiten von Rechtsanwälten, Richtern und Verwaltungsjuristen zu unterscheiden. Im Anschluß daran werden die Einsatzmöglichkeiten des Computers in der juristischen Ausbildung dargestellt.



1. Rechtsanwalt

Von den 115.000 am 1.1.1986 in der Bundesrepublik erwerbstätigen Juristen arbeiteten 48.700 als Rechtsanwälte, 35.000 als Verwaltungsjuristen bei Bund, Ländern und Kommunen, 21.000 in der Justiz und 12.000 in der Wirtschaft. Vor diesem Hintergrund scheint es mir gerechtfertigt, zunächst das Tätigkeitsfeld des Rechtsanwalts ins Auge zu fassen.
Die knapp 50.000 Rechtsanwälte in der Bundesrepublik arbeiten in ca. 23.000 Rechtsanwaltskanzleien. Davon sind nach Schätzungen von Experten gegenwärtig weniger als 2000 Kanzleien zumindest teilweise mit Computern ausgestattet (1). Bei den Computern überwiegen - bis auf wenige große Kanzleien - Lösungen mit Personalcomputern. Die Zahl der angebotenen Rechtsanwaltsprogramme liegt bei ca. 23. Davon sind ca. 14 integrierte Programme. Unter integrierten Programmen versteht man Softwarepakete, die modulweise die wesentlichen Arbeitsbereiche eines Anwaltsbüros abdecken und Schnittstellen zur Datenkommunikation zwischen den einzelnen Modulen enthalten.
Für den Atari-ST-Standard gibt es derzeit noch kein integriertes Anwaltsprogramm. Von Joachim Fiedler werden derzeit folgende Einzelprogramme vertrieben: Programm zur Erstellung von Mahnbescheidsantägen (MBxx. PRG Vers. 1.3),in der Version 1.3s für Vordrucke der Hans-Soldan-Stiftung, 99.- DM; Textmuster - Ordner für 1ST-Word bzw. Wordplus in den Bereichen Prozeß-, Unfall-, Familien- und Mietrecht, 49,- DM; BRAGO- und Gerichtskostentabelle als ACC-Programm (BRAGOGK.ACC), gegen Diskette zzgl. 7 DM,- (2).

geht weiter...

MEGA /// TEAM Computersysteme

Qualität ist unsere Stärke

MEGA WORK ST

Software zum Arbeiten, Überflüssiges stört nicht

★ **Fakturierung**
★ **Adressverwaltung**

- Maus- oder Tastaturbedienung möglich
- 10000/1000 Adressen je Datei (Festplatte/Diskette)
- Suchen über Adress Nr., oder sequentiell über Felder
- Adressen als ASCII-Datei ausgebbar
- Bis 6000/1000 Artikel je Datei (Festplatte/Diskette)

★ **Serienbrieferstellung**
★ **Lagerverwaltung**

- Artikel eingeben, ausgeben, ändern, sortieren, löschen
- Serienbrieferstellung über individuellen Texteditor
- Variabler Etikettendruck (Adressen oder Kundendaten)
- Rechnung/Angebot/Lieferschein-/Bon vorgegeben, weitere Formulare können einfach selbst erstellt werden.

★ **Textverarbeitung**
★ **Relationale Datenbank**

- Alle Kalkulationsmöglichkeiten im Programm integriert
- Ausdruck von Lagerbestand, Kundenkartei, Bestell-Liste...
- MEGAwork läuft auf allen Atari ST Computern,
- auch mit Farbmonitor

198,—

Hardware für ST und PC

ATARI SH 205	1298.00
Vortex HDplus 20	1248.00
Vortex HDplus 40	2498.00
Brother M 1209	748.00
Brother M 1409	998.00
Brother M 1724L	1848.00
Brother HL-8 LaserAs	5998.00
NEC 2200	998.00
NEC P 6	1198.00
NEC P 7	1598.00
EPSON LQ 500	998.00
EPSON LQ 850	1398.00
Handy-Scanner für ST	889.00
512 KB Erweiterung	229.00
Echtzeituhr	109.00
Monitor-Umschaltbox	55.00
FL-1 Adapter	45.00
Centronics-Kabel	25.00
3.5" Reinigungsset	12.50
5.25" Reinigungsset	10.00
NEC 1036A Laufwerk	219.00

Software für ATARI ST und PC

Text- und DTP-Programme

SIGNUM!Zwei	389.00
STEVE incl. Lehrbuch	329.00
1st Word plus 2.02	198.00
StarWriter ST	198.00
MEGAtext ST	198.00
WordPerfect ST	798.00
Timeworks Publisher	369.00
Calamus junior	298.00
Calamus Professional	948.00
Publishing Partner	198.00
BECKERpage	395.00

Grafik

Easy Draw	229.00
Art Director, deutsch	119.00
Film Director, deutsch	119.00
IMAGIC	489.00
STAD 2.0	159.00
GFA ARTIST	139.00
GFA MOVIE	139.00
Monostar Plus	139.00
MEGAetikett	49.00
CAD 3 - D	139.00
StarPainter	98.00

Spiele für ST

Emerald Mine	29.00
Gokart Grand Prix	29.00
Space Pilot	29.00
Terra Nova	29.00
Airball	49.00
Thyphoon	49.00
Tass Times in Tonetown	59.00
Indiana Jones	79.00
Marble Madness	79.00
The Sentinel	79.00

Kfm. Programme

ZAPP ST	198.00
VIP	698.00
ST-TOS-Manager	98.00
ST-Kundenverwaltung	145.00
ST-Lohn und Gehalt	189.00
ST-Faktura/Lagerverw.	369.00
ST-Finanzbuchhaltung	469.00
BECKERbase ST	98.00
BECKERtools ST	98.00
Hausverwaltung ST	469.00
MEGAwork ST	198.00

Weitere Software für ST und PC

Megamax Modula-2	389.00
Megamax C	389.00
Das Harddisk Utility	65.00
FlexDisk	65.00
G Roboterarm	269.00
AS Sound Sampl.II Softw.	145.00
AS Sound Sampl.II Hardw.	245.00
G COPY	95.00
G FAST SPEEDER	125.00
G RAMDISK II	48.00
G INTERPRINT II o. Ramd.	48.00
G INTERPRINT II m. Ramd.	95.00
G Harddisk Help & extens.	125.00
G DISK HELP	75.00
G DISKMON II	95.00
G SCANNER	289.00
G Logik Analyser	429.00
G GEM Retrace Recorder	95.00
G Sympatic Paint	295.00
OMIKRON Basic auf Disk	169.00
OMIKRON Compiler	169.00
OMIKRON MIDI-Library	75.00
OMIKRON D.R.A.W.	75.00
OMIKRON IDEAL	75.00
Superbase	239.00
acta ST	78.00
Multi Hardcopy	99.00
Laser Brain	289.00
Laser de Luxe	239.00
Spekulieren und Gewinnen	689.00
LOGISTIX	389.00
GFA-Basic 2.02	95.00
GFA-Compiler 2.02	95.00
GFA-Basic 3.0	198.00
GFA-Basic 68881	329.00
GFA-Objekt	189.00
GFA-Starter	58.00
GFA-Draft plus	339.00
Lattice C 3.04	279.00
1st Freezer	139.00
1st Speeder	89.00
Dizzy Wizard	65.00
StarWriter PC	389.00
GFA-Desk PC	95.00
GFA-Fakt PC	139.00
GFA-Draft plus PC	189.00

MEGA /// TEAM Computersysteme

Kirchhellener Str. 262 · 4250 Bottrop · Tel. (02041) 94842

Am 5. März 1988 findet unsere 2. große ATARI-Fete statt. Zahlreiche Aktionen (ausführliche DTP-Vorstellungen, Sound Sampling Show etc.) machen Ihren Besuch zu einem besonderen Erlebnis.

Und. Sie können jedes Hard- und Softwareprodukt bei uns ausgiebig testen. Wir möchten Sie noch einmal auf unsere günstigen Finanzierungsangebote hinweisen, die schnell, günstig und – für beide Seiten – effektiv sind.

Wir sind autorisierte Vertragshändler von:
ATARI, Brother, Vortex, DATA BECKER, Kieckbusch, Sybex, G DATA, OMIKRON MGT Software, Q-Line, Markt + Technik

fordern Sie unseren umfangreichen Katalog an

Laden-Geschäftszeiten
Mo. - Fr.
9.00 - 13.00 Uhr
14.30 - 18.30 Uhr
Samstag
9.00 - 14.00 Uhr
Langer Samstag
9.00 - 18.00 Uhr

☐ Senden Sie mir bitte Ihren Katalog (1,— DM in Briefmarken liegt bei)
☐ Hiermit bestelle ich per Nachnahme: (incl. kostenlosem Katalog)

Vorname, Name
Straße, Hausnummer
PLZ, Ort
Telefonnummer
Datum, Unterschrift

Kurz vor der Fertigstellung soll die Datenbank KANZLEI stehen, bei der es sich um eine ADIMENS ST-Applikation handelt. Es soll sich dabei um eine "Mandanten- und Prozeßverwaltung mit Gebührenabrechnung" handeln. Der Preis soll 149,- DM betragen (3).

Auf beide Programme wird in einem der nächsten Hefte ausführlich eingegangen werden. Wenn eines der Programme zu diesen Preisen mit herkömmlichen Programmen vergleichbare Leistungen bieten könnte, müßte es nach menschlichem Ermessen ein Renner werden. Rechtsanwälte sind äußerst preisbewußte Investoren. So ist der Markt für Rechtsanwaltsprogramme sehr schwierig. Die Anbieter für Rechtsanwaltssoftware dürfen sich zu den Marktführern rechnen, wenn sie über 200 Installationen vorweisen können. Dies macht deutlich, daß der eigentliche Durchbruch der elektronischen Datenverarbeitung im Rechtsanwaltsbüro noch bevorsteht. Es ist müßig, in diesem Zusammenhang zu fragen, ob diese Branche im Vergleich zu anderen freien Berufen "rückständig" ist. Neben den auf der subjektiven Ebene angesiedelten Vorbehalten gerade älterer Rechtsanwälte und Mitarbeiter in Anwaltsbüros (sog. "Akzeptanzproblem") wird die Entscheidung über den Einsatz von Computern letztlich von betriebswirtschaftlichen Kosten-/Nutzenanalysen abhängen. Der Einsatz von Computern ist für den Rechtsanwalt zunächst eine Rationalisierungsinvestition. Während sich die Kosten relativ genau beziffern lassen, nämlich für die Geräte, also den Rechner, Eingabegeräte (Tastatur, evtl. "Maus"), Ausgabegeräte (Drucker), externe Speicher (Festplatten etc.) und weitere Peripheriegeräte (Monitor, Akkustikkoppler, Modem) (=sog. Hardware), das Programm (sog. Software), Wartungs- und Schulungskosten, Postgebühren bei Datenfernübertragung, Gebühren für Mailbox oder Jurisanschluß, ist der Rationalisierungsgewinn nicht exakt abschätzbar (Einsparung zusätzlicher oder Entlassung bisheriger Mitarbeiter, mehr Freizeit oder größerer Mandantenstamm des Anwalts, Einsparung externer Buchhaltungs- und Steuerberatungskosten), vorallem, wenn zugleich eine Verbesserung der Qualität der juristischen Arbeit angestrebt wird.

Angesichts stagnierender Umsätze der Anwälte und immer günstigerer und leistunsfähigerer Hardware wird der Einsatz von EDV in den nächsten Jahren zunehmend auch für mittlere und kleine Anwaltskanzleien interessant werden.

Zudem werden Berufsanfänger, die weder im öffentlichen Dienst noch in einer Anwaltskanzlei eine Anstellung finden, sich als Rechtsanwalt selbständig machen - schon um ihre juristische Qualifikation zu erhalten und Berufserfahrung zu sammeln. Die Ertragslage neugegründeter Rechtsanwaltsbüros wird es zumindest in der Anfangsphase kaum zulassen, Büroangestellte zu entlohnen, so daß deren Aufgaben vom Rechtsanwalt miterledigt werden müssen. Der Einsatz elektronischer Datenverarbeitung kann hier mit dazu beitragen, die Ergebnisse der Arbeit dennoch professionell aussehen zu lassen.

Integrierte Anwaltssoftware unterstützt den Betrieb einer Anwaltskanzlei gegenwärtig in folgenden Bereichen:

a. Die Aktenverwaltung

Die Basis jedes integrierten Rechtsanwaltsprogrammes ist die Mandanten- und Aktenverwaltung. Bei der Aufnahme eines neuen Mandates wird der Akte eine in der Regel automatisch fortlaufende Registernummer zugeteilt, der die Stammdaten des Verfahrens zugeordnet werden. Die Registernummer dient der Verknüpfung der einzelnen Module des Anwaltsprogrammes untereinander. Von dort aus kann dann rasch auf die Stammdaten zugegriffen werden. Der Organisation des Betriebsablaufes dient zusätzlich eine mehr oder weniger ausgefeilte Terminverwaltung, die Mandanten- und Gerichtstermine, Fristen und Wiedervorlagen speichert.



b. Anwaltliche Routinetätigkeit

Zur Unterstützung anwaltlicher Routinetätigkeiten, die im wesentlichen nach dem gleichen Schema, insbesondere mit fast gleichlautenden Schreiben, ablaufen, gibt es besondere Programmteile, die aus vorformulierten Texten mit variablen Platzhaltern bestehen. Nach Eingabe der Registernummer werden diese Platzhalter aus den Stammdaten der Aktenverwaltung und der Buchhaltung aufgefüllt. Durch die Verknüpfung mit dem Terminkalender ist eine sog. "Stapelverarbeitung" möglich. In diesem Fall wird beim Erreichen eines vorherbestimmten Termines und vorher festgelegter Umstände (beispw. Nicht- oder nur Teilzahlung des Gegners) der nächste Verfahrensschritt durch Ausdruck des entsprechenden Textes (beispw. Antrag auf Erlaß eines Mahnbescheides) eingeleitet. In der Praxis werden solche "Routineprogramme" für die Bereiche Mahn- und Zwangsvollstreckung und Verkehrsunfallbearbeitung angeboten.

c. Textbe- und -verarbeitung

Die Textverarbeitung ist sicherlich die häufigste Anwendung im Anwaltsbüro. Sie läßt sich wiederum unterteilen in die Erstellung von Texten, die für den Einzelfall formuliert wurden (beispw. Klageschrift) und die Arbeit mit vorformulierten Textbausteinen (4), die ebenfalls variable Platzhalter enthalten können. Derartige vorformulierte Textblöcke ersetzen u.a. die in allen Anwaltspraxen geläufigen sog. "Rotzettel".

Natürlich sind auch Mischformen zwischen beiden Varianten möglich. Beispielsweise kann der Aufbau einer Kündigungsschutzklage durch ein Skelett aus Textbausteinen, die z.B. rechtliche Ausführungen zur Beteiligung des Betriebsrats bei einer Kündigung nach _ 102 BetrVG oder zu den Grundsätzen der Sozialauswahl (_ 1 Abs. 3 KSchG) enthalten, strukturiert werden. Die besonderen Umstände des Einzelfalles können dann mit individuellen Formulierungen unter diese "Obersätze" subsumiert werden.

geht weiter...

SILVER REED SCANNER PRINTER SPAT



Der Silver Reed Scanner Printer SPAT macht aus Ihrem ATARI ST und MEGA ST ein Grafik-Atelier.
Sie übertragen damit Bilder auf Ihren Bildschirm. Sekundenschnell. Diese können Sie dann ganz nach Wunsch verändern oder neu gestalten.
Vergrößern, verkleinern, Dinge wegnehmen oder hinzufügen, Schrift einfügen und, und, und …
Über den SPAT drucken Sie Ihr Kunstwerk aus – auf Thermopapier. Oder … in Top-Qualität über Ihren Matrix- oder Laserdrucker.

Aber der Silver Reed Scanner Printer SPAT hat noch weitere Vorteile:

- Kopieren – dabei ist die Kopiergröße je nach Vorlage einstellbar.
- Wartungsfrei.
- Kein Verbrauchsmaterial (kein Toner, kein Entwickler, keine Trommel).

Für nur DM 1.995,–
unverbindl. Preisempfehlung (incl. gesetzl. MWSt)

COUPON
Bitte senden Sie mir mehr Information:

Name ____________ Straße ____________

PLZ/Ort ____________ Telefon ____________

Coupon bitte senden an:
Silver Reed International GmbH · Nordendstraße 48 · 6082 Mörfelden-Walldorf

d. Rechentätigkeiten

Beim Rechnen ist der Computer in seinem Element. In engerem Sinne juristische Rechenprogramme, die als Standardsoftware vertrieben werden, finden sich nur im Bereich des Unterhaltsrechts und - soweit ersichtlich - auch nur als Zusatzmodule, so daß nicht auf bereits anderweitig gespeicherte Daten zurückgegriffen werden kann.
Wesentlich ausgefeilter sind dagegen regelmäßig die Buchhaltungsprogramme. Rechtsanwälte brauchen nur eine einfache Einnahme/Überschuß-Buchführung zu betreiben. Diese kann durch die Computerunterstützung im Hause betrieben werden, so daß die mitunter erheblichen Kosten für eine externe Buchhaltung gespart werden können, zumal auch im letzten Falle die notwendigen Belege systematisch geordnet werden müssen. Die hier liegenden Einsparungsmöglichkeiten, die nur mit geringem eigenen Mehraufwand verbunden sind, tragen erheblich dazu bei, daß sich die Einführung der EDV im Anwaltsbüro amortisiert. Mit der Finanzbuchhaltung ist regelmäßig die Mandantenbuchhaltung verknüpft, in der die Buchungen nach Mandanten spezifiziert werden. Die Mandantenbuchhaltung wiederum liefert die Daten für die computerunterstützte Honorar- und Kostenabrechnung.

e. Statistische Auswertungen

Die computerunterstützte Buchhaltung ermöglicht zugleich eine relativ schnelle betriebswirtschaftliche Auswertung, die schon wegen der Umsatzsteuer regelmäßig monatlich vorgenommen wird. Die betriebswirtschaftliche Auswertung ist nicht nur in der Lage, die Umsatz-, Kosten- und Ertragsentwicklung innerhalb eines Zeitraumes transparent zu machen. Je nach Zuordnung der Registernummer zu Sachbearbeitern oder Referaten ist auch eine Feinanalyse möglich. Von dieser Möglichkeit wird jedoch - gerade bei Sozietäten - in sehr unterschiedlichem Maße Gebrauch gemacht.



f. Vertragsgestaltung

Module für den Notariatsbereich sind erst in jüngster Zeit auf den Markt gekommen. Auch hier geht es einmal um rein organisatorische Aufgaben, nämlich die Verwaltung der laufenden und abgeschlossenen Vorgänge in der Urkundenrolle.
Zum anderen bietet es sich an, Textblöcke und Textbausteine für den Schriftverkehr und für häufig wiederkehrende Vertragsgestaltungen (z.B. Grundstückskaufvertrag) zu speichern (5).
Das Vorhandensein und die Qualität dieses Programmteiles kann als Indikator für die Qualität des Gesamtprogrammes herangezogen werden, weil hier deutlich wird, in welchem Ausmaß der Anbieter über die allgemein üblichen Standardanwendungen hinaus auf spezifisch juristische Bedürfnisse eingeht.

g. Datenbank

Die meisten Anwaltsprogramme enthalten als Option auch ein Datenverwaltungsprogramm, in dem der Rechtsanwalt Rechtsprechung und Literatur entsprechend seinem jeweiligen Anwendungsgebiet speichern und verwalten können soll. Diese Module leiden regelmäßig unter drei Mängeln: Zum einen sind die Eingabefelder ("Masken") nicht auf juristische Besonderheiten abgestimmt. Das gilt insbesondere für Rechtsprechungsnachweise einschließlich der Zuordnung mehrerer Instanzen bzw. Fundstellen. Zum zweiten gibt es wenige Programme, die sowohl eine Verschlagwortung der Dokumente zulassen und darüberhinaus auch eine Volltextrecherche zulassen (6). Schließlich fehlt entweder eine Übergabemöglichkeit an die im Anwaltsprogramm enthaltene Textverarbeitung, so daß weder der Nachweis selbst noch in Verbindung damit gespeicherte Exzerpte, Leitsätze oder anderer Text in mit der Textverarbeitung erstellte Texte direkt übernommen werden können, oder die Übernahme ist nur als ASCII-Datei möglich (was regelmäßig eine "Nachbehandlung" des übernommenen Textes nötig macht) (7). Als einfacher "Karteikasten" zum Nachweis von Fundstellen sind diese Optionen aber zu teuer.
Im übrigen muß die Literatur und Rechtsprechung erst einmal in der Datenbank gespeichert werden. Es ist verwunderlich, daß bisher noch kein Anbieter auf die Idee gekommen ist, in regelmäßigen Abständen eine Literatur- und Rechtsprechungsübersicht - und sei es auch nur zu bestimmten Rechtsgebieten - zu vertreiben. Wieviel Arbeit könnte gespart werden, wenn beispielsweise die NJW-Leitsatzkartei (8) auf Diskette erhältlich wäre, von den erheblich differenzierteren Suchmöglichkeiten einmal abgesehen.
Einen ersten Schritt in diese Richtung bildet die Fundstellendatei "BGH-DAT", die von Richtern des BGH herausgegeben und vom Carl Heymanns Verlag vertrieben wird. Damit wird jedoch fast ausschließlich die Rechtsprechung des BGH in Zivilsachen dokumentiert. In Relation zu diesem beschränkten Umfang erscheinen die Kosten für diese Datenbank zunächst einmal als zu hoch (9).
Einen zweiten hoffnungsvollen Schritt - ebenfalls voerst nur für den Atari ST-Standard - bildet die Fundstellendatei "WEHRDAT" (10). Auf beide Programme wird in einem der folgenden Beiträge detailliert eingegangen.

2. Richter und Staatsanwälte

Im Justizbereich lassen sich mehrere typische Einsatzgebiete der elektronischen Datenverarbeitung ausmachen: Die Unterstützung des Geschäftsablaufes bei den Geschäftsstellen, die Bearbeitung ursprünglich richterlicher Routinetätigkeiten, die weitgehend an die Rechtspfleger delegiert wurde, die Textbe- und -verarbeitung, Rechentätigkeiten und statistische Zwecke.

a. Die Unterstützung des Geschäftsablaufes bei den Geschäftsstellen

Das Institut für Datenverarbeitung im Rechtswesen der Gesellschaft für Datenverarbeitung und Mathematik mbH Bonn (GMD) hatte 1976 ein Konzept zur Unterstützung des gerichtlichen Geschäftsbetriebes durch den Einsatz elektronischer Datenverarbeitung entwickelt, das zuerst am Amtsgericht Wiesbaden und am Landgericht Frankfurt (später auch dort am Amtsgericht) erprobt wurde. Ab 1979 übernahm das Hessische Justizministerium die Pilotanwendungen. Das Programm AUGe (=Automationsunterstützung im gerichtlichen Geschäftsbetrieb) dient zur Verwaltung der anhängigen Verfahren (Verfahrenseingang, -änderung und -ergänzung, Aktenkontrolle, Fristen- und Terminverwaltung sowie der Erstellung von Ladungen, Versäumnisurteilen und Kostenrechnungen. Außerdem wird der

geht weiter...

Verfahrensabschluß erfaßt und der Zählkartendatensatz erstellt. Die Leitungsebene der Gerichte hat einen jederzeitigen Zugriff auf aktuelle Daten über Geschäftsanfall und Erledigungen. In eine ähnliche Richtung zielen die Arbeiten an dem Programm SOJUS (=Software für Geschäftsstellen und Kanzleien in der Justiz). Die computerunterstützte Geschäftsstellenverwaltung ist heute nicht mehr auf die Zivilgerichtsbarkeit beschränkt, auch Straf- und Verwaltungsgerichte aller Instanzen sowie die Staatsanwaltschaften machen von ihr Gebrauch. In beiden Fällen handelt es sich um Anlagen der mittleren Datentechnik. Für den Richter oder den Staatsanwalt ist diese Entwicklung nur deswegen von Bedeutung, weil die das gerichtliche Verfahren betreffenden Verfügungen von ihm in "computergerechter" Form getroffen werden müssen.

b. Bearbeitung ursprünglich richterlicher Routinetätigkeiten

Baden-Württemberg hat als erstes Bundesland das automatisierte Mahnverfahren gem. _ 689 ZPO eingeführt. Auch in anderen Bundesländern wurden die Verfahren zur Abgabe der eidesstattlichen Versicherung und die Führung des Schuldnerverzeichnisses auf elektronische Datenverarbeitung umgestellt. Diese Anwendungen sind mit einer EDV-unterstützten Namenskartei beim Registergericht verbunden und fertigen Ausdrucke für die Industrie- und Handelskammern. Darüberhinaus gibt es in unterschiedlichem Ausmaß Anwendungen in Grundbuchsachen, in Zwangsversteigerungs- und Konkursverfahren, also in Bereichen, die großenteils gem. _ 3 RPflG dem Rechtspfleger übertragen sind, sowie in Landwirtschaftssachen und in Familiensachen. So gibt es beispielsweise Verfahren für die computergestützte Festsetzung des Regelunterhaltes (__ 642a ff. ZPO) und die Unterhaltsabänderung nach _ 641 ZPO. Stellenweise sind diese Anwendungen mit den vorstehend beschriebenen Verfahren zum computergestützten Geschäftsstellenablauf verbunden (11).

c. Textbe- und -verarbeitung

An den Landgerichten sowie an Verwaltungsgerichten und Oberverwaltungsgerichten werden häufig Textverarbeitungsautomaten eingesetzt. Beispielhaft ist auch das Arbeitsgericht Ludwigsburg, wo die elektronische Textverarbeitung in der Rechtsantragstelle, bei richterlichen Verfügungen und Anschreiben sowie für die Protokollierung der Güte- und Kammerverhandlung eingesetzt wird. Bei einigen Gerichten wird auch von der Möglichkeit Gebrauch gemacht, bei der Absetzung von Urteilen auf gespeicherte Textbausteine zurückzugreifen; dies gilt insbesondere für Asylverfahren, Numerus-clausus-Verfahren, in Familien- und Verkehrsstrafsachen. Grundsätzlich ist die Verwendung von Textbausteinen für die Abfassung von Urteilen zulässig. In der von den Richtern unterzeichneten Urschrift müssen jedoch sämtliche Textbausteine entsprechend der gewünschten Endfassung ausformuliert sein.

Die elektronische Datenverarbeitung ermöglicht über das reine Abrufen von Textbausteinen hinaus ihre Gliederung und Sortierung nach jedem erdenklichen Gesichtspunkt. Als vergleichsweise einfache Anwendung kommt die Speicherung und Gliederung eines Sach-verhaltes in einer Textdatei unter den Punkten "Unstreitiges/ Klägervortrag/Beklagtenvortrag" in Betracht. Diesem Sachverhaltsteil mit Bezugnahmen auf die Gerichtsakte kann zugleich ein Bewertungsteil in Form der Entscheidungsgründe angehängt werden. Notwendige Ergänzungen und Korrekturen, die sich aus dem weiteren Schriftwechsel der Parteien oder der mündlichen Verhandlung ergeben, können problemlos eingefügt werden.

Bietet die elektronische Datenverarbeitung schon bei der Aufarbeitung eines "normalen" Sachverhaltes und seiner rechtlichen Bewertung Vorteile durch Zeitersparnis und bessere Durchdringung des Stoffes, so gilt dies umso mehr bei komplexen Großverfahren. In der juristischen Literatur ist dies insbesondere für Wirtschaftsstrafverfahren aus der Sicht des Staatsanwaltes dokumentiert (12). Hier wird die Synthese zwischen den Möglichkeiten elektronischer Datenverarbeitung für die juristische Arbeit noch deutlicher: In einem ersten Arbeitsschritt wird mit Hilfe eines Datenbankprogrammes eine Aktendokumentation angelegt. Dabei geht es zunächst nur um die Erfassung des Sachverhaltes im Hinblick auf ein etwaiges strafbares Verhalten. In einem zweiten Schritt wird der Sachverhalt nach juristischen Kriterien aufgearbeitet. In einem ersten Teilschritt erfolgt die Gliederung des Textes, wobei es beliebig ist, ob es sich um eine Anklageschrift, eine Einstellungsverfügung oder ein Urteil handelt. In einem zweiten Teilschritt wird diese Gliederung aus der Aktendokumentation aufgefüllt. Das Ziel der Zeitersparnis und der besseren Durchdringung des Sachverhaltes wird einmal dadurch erreicht, daß die gespeicherten Daten nach den verschiedensten Kriterien (Täter, Opfer, Zeitraum der Handlung, Schadenshöhe) selektiert werden können. Die Benutzung sog. "integrierter" Software, also miteinander verbundener und austauschfähiger Einzelprogramme wie z.B. Datenverwaltung, Textverarbeitung, Tabellenkalkulation, graphische Darstellungen von Zahlen gestattet den reibungslosen Transfer von Daten aus der Datenbank in die Textverarbeitung. Rechenaufgaben können von der Tabellenkalkulation wahrgenommen werden; das Ergebnis kann schließlich durch Torten- oder Balkendiagramme graphisch dargestellt werden.

Für diese Tätigkeiten wären sicherlich auch Atari ST-Computer mit entsprechenden Anwendungsprogrammen einsetzbar. Für die Textverarbeitung in der Justiz wäre - ebenso wie in größeren Anwaltskanzleien - die Netzwerkfähigkeit eine unabdingbare Voraussetzung. Hierüber gibt es aber noch keine publizierten Praxiserfahrungen. Gleiches gilt über den Einsatz "integrierter" Programme wie VIP Professional an Einzelarbeitsplätzen (13).

d. Rechentätigkeiten

Richter haben auch schon vor der Erfindung des Computers rechnen müssen. Durch die elektronische Datenverarbeitung werden ihnen jedoch auch komplizierteste Berechnungen leicht gemacht. Das gilt nicht nur für die Berechnung von Unterhaltsansprüchen und des Versorgungsausgleiches, sondern beispielsweise auch für die Zinsvergleiche bei der Prüfung (sittenwidriger) Konsumentenratenkredite über die Quotenberechnung im Konkursverfahren bis hin zum Kostenrecht ("Baumbachsche Formel"). Soweit ersichtlich, existieren entsprechende Programme bislang nur für MS-DOS-Computer (14).

geht weiter...

EINEN BLICK



G SCANNER

Problemloses Digitalisieren von s/w-Bildern mit Ihrem Drucker ermöglicht der G SCANNER. Die hochwertige Abtastdiode von HP garantiert die ausgezeichnete Qualität von 200 dpi. Sie wird einfach auf dem Druckknopf Ihres Druckers befestigt. Mitgeliefert wird Software, die DEGAS und Normalbilder erstellt und Bildeditieren ermöglicht. Der G SCANNER ist voll kompatibel zu STAD und SYMPATIC PAINT (Scannoption).

DM 298,—

P6 Halterung für Diode **DM 35,—**



Roboterarm

Der semiprofessionelle 5achsige Roboterarm wird anschlußfertig für ST, AMIGA oder PC/AT geliefert. Für Schulungszwecke, Lehr- und Demonstrationszwecke oder einfach zum Spielen bestens geeignet. Technische Daten: Höhe ca. 54 cm, industriegelb, ca. 2,5 kg schwer, einfacher Anschluß an Drukkerport. Auch über Joysticks zu steuern.

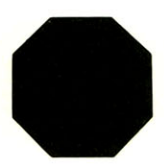
Anschlußfertig **DM 269,—**
Spezialnetzteil **DM 99,—**
Batteriesatz **DM 10,—**



TV Modulatoren

Schließen Sie Ihren ST an den Farbfernseher an, und sparen Sie sich einen teuren RGB Monitor. Unsere Modulatoren bringen ein scharfes Bild mit allen Mischtönen und einen Tonausgang extern und einen über den Fernseher. Das heißt: Bestes Bild und bester Tor für alle Fernseher.

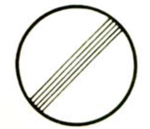
Typ A mit Monitorumschalter (ATARI-Fernseher) **DM 249,—**
Typ B ohne Monitorumschalter **DM 239,—**
Typ C FBAS Wandler für Geräte mit AV oder FBAS Eingang **DM 169,—**



RAM Erweiterungen

Unsere Ram Erweiterungen sind völlig neu. Sie werden ohne Lötarbeiten im Gerät gesteckt. Keine Leiterbahntrennungen – keine Treibersoftware. Die Platinen werden unbestückt geliefert und können mit 256 KBit Chips oder mit 1 MBit Chips bestückt werden. So erreicht man – je nach Wunsch – eine Erweiterung um 512 KB, 2 MB oder 4 MB. Dazu benötigen Sie nur eine Platine. Das bedeutet: zukunftssicher und problemlos.

für 520, 520+, 260, STM **DM 249,—**
für 1040 STF **DM 249,—**

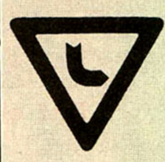


LOGIC ANALYSER

Endlich ein Logik Analyser, den sich jedermann leisten kann. Dabei können wir auf die Leistungsmerkmale von 5-10fach teureren Geräten hinweisen: 16 Kanäle bis 600 KHz, komfortable GEM Software, verschiedene Triggerungsmöglichkeiten usw. Das Gerät wird auf den Romport gesteckt und anschlußfertig mit Software geliefert.

DM 448,—

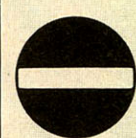
NEUHEITEN



SYMPATIC PAINT

Der ATARI ST als Grafik Workstation der Superlative! Alle Funktionen, die für ein effektives und komfortables Zeichnen, Malen und Konstruieren erforderlich sind, wurden integriert. Der Benutzer hat nicht nur die Möglichkeit der Erstellung einer individuellen Grafiktoolbox, sondern auch eine Animationsselektion, mit der Bilder laufen lernen. Bilder können dabei mit dem G SCANNER und EASYTIZER eingelesen werden. Selbstverständlich kann die laufende Animation mit Sound unterlegt werden. Dabei können nicht nur alle ATARI Sound Register gezogen werden, sondern auch digitaler Sound vom AS SOUND SAMPLER II eingebaut werden. Trickfilme mit digitalem Sound! Weitere Informationen in diesem Heft.

DM 298,—



GEM Retrace Recorder

Der Knüller! Wie von Geisterhand bewegt sich die Maus, laufen Kommandos ab, werden Aktionen gestartet. Der GEM Retrace Recorder ermöglicht es, jegliche Aktion (Mausbewegung, Menüs, Klicks, Tastatur usw.) aufzuzeichnen, und auf Tastendruck jederzeit original zu wiederholen. Das heißt: GEM Makros erstellen für Aktionen, die ständig wiederholt werden (auch in jedem Programm); super Auto Boots erstellen; selbstlaufende Demos von jedem Programm ohne Programmeingriffe und Kenntnisse u.v.m. Die Anwendungsmöglichkeiten sind fast unbegrenzt!

DM 99,—



Hardware Uhr

Die Uhr wird ohne Lötarbeiten im Rechner eingebaut, so daß der ROM Port frei bleibt. Sie haben ständig die aktuelle Zeit und das aktuelle Datum! Die Lithium Batterie garantiert hohe Ganggenauigkeit und 10 Jahre Laufzeit bei voller Schaltjahrerkennung. Dr sensationelle Preis für die anschlußfertige Uhr beträgt nur

DM 79,—

G DATA

Siemensstraße 16 4630 Bochum 1
Telefonische Bestellungen:
023 25 / 6 08 97

Schweiz: Senn Computer AG
Langstr. 31
CH-8021 Zürich

Österreich: Computershop Rittner
Hauptstr. 34
A-7000 Eisenstadt

BENELUX:

G DATA BeNeLux
Postbus 70
NL-2000 AB
Haarlem
Tel. 023 / 32 13 31

COUPON

Hiermit bestelle ich ☐ Informationsmaterial (DM 2,— beiliegend)
☐ 2 Demodisketten und Informaterial (DM 10,— beiliegend)

☐ per Nachnahme (ca. DM 6,50 Versandkosten)
☐ Verrechnungsscheck liegt bei (+ DM 5,— Versandkosten)

Absender:

e. Statistische Zwecke

Ganz andere Zwecke verfolgt dagegen das Projekt JUSTIS (=Justizstatistik-Informationssystem). Neben den Zählkartendaten werden "Personalstatistiken und justizrelevante Sekundärdaten aus dem Umfeld der Gerichte" in JUSTIS gespeichert. JUSTIS gibt u.a. Auskunft über den Verfahrensinhalt von Prozessen, über die Kategorie der Parteien, die Anzahl der Termine mit oder ohne Beweisaufnahme und über die Gerichtskosteneinheit. Auch hier handelt es sich um ein Vorhaben auf der Ebene der mittleren Datentechnik, das in diesem Zusammenhang nicht weiter interessiert.

f. Datenbanken

Hier gilt das bereits beim Arbeitsfeld des Rechtsanwalts Ausgeführte entsprechend.

3. Verwaltung

Auf den EDV-Einsatz im Verwaltungsbereich einzugehen, würde den Rahmen der vorliegenden Abhandlung sprengen, zumal es sich in der Regel um zentralisierte Datenverarbeitung handelt. An dieser Stelle sei daher nur darauf hingewiesen, daß die EDV in der Verwaltung - ebenso wie im privatwirtschaftlichen Bereich - in unterschiedlich starkem Maße eingesetzt wird. Hauptanwendungsgebiet ist hier weniger die Unterstützung der juristischen Arbeit im engeren Sinne, sondern vorwiegend die des Geschäftsablaufes durch Speicherung personenbezogener Stammdaten in der Leistungsverwaltung. Am weitesten fortgeschritten ist der EDV-Einsatz naturgemäß im Registerbereich (beispw. Bundeszentralregister, Einwohnermeldeämter) und in der Abwicklung des Zahlungsverkehrs insbes. durch Rechenzentren (beispw. Steuer- und Gebührenbescheide, Landesämter für Besoldung und Versorgung).



4. Kommunikationsmöglichkeiten zwischen den einzelnen Arbeitsfeldern und der Zugriff auf externe Datenbanken

Eine Betrachtung der Unterstützungsmöglichkeiten elektronischer Datenverarbeitung für die juristische Arbeit wäre unvollständig, wenn nicht auch die neuen Kommunikationsmöglichkeiten betrachtet würden.

Hier auf Einzelheiten eingehen zu wollen, erscheint mir müßig, da dieses Thema auch für Juristen bereits umfassend aufgearbeitet ist (15).

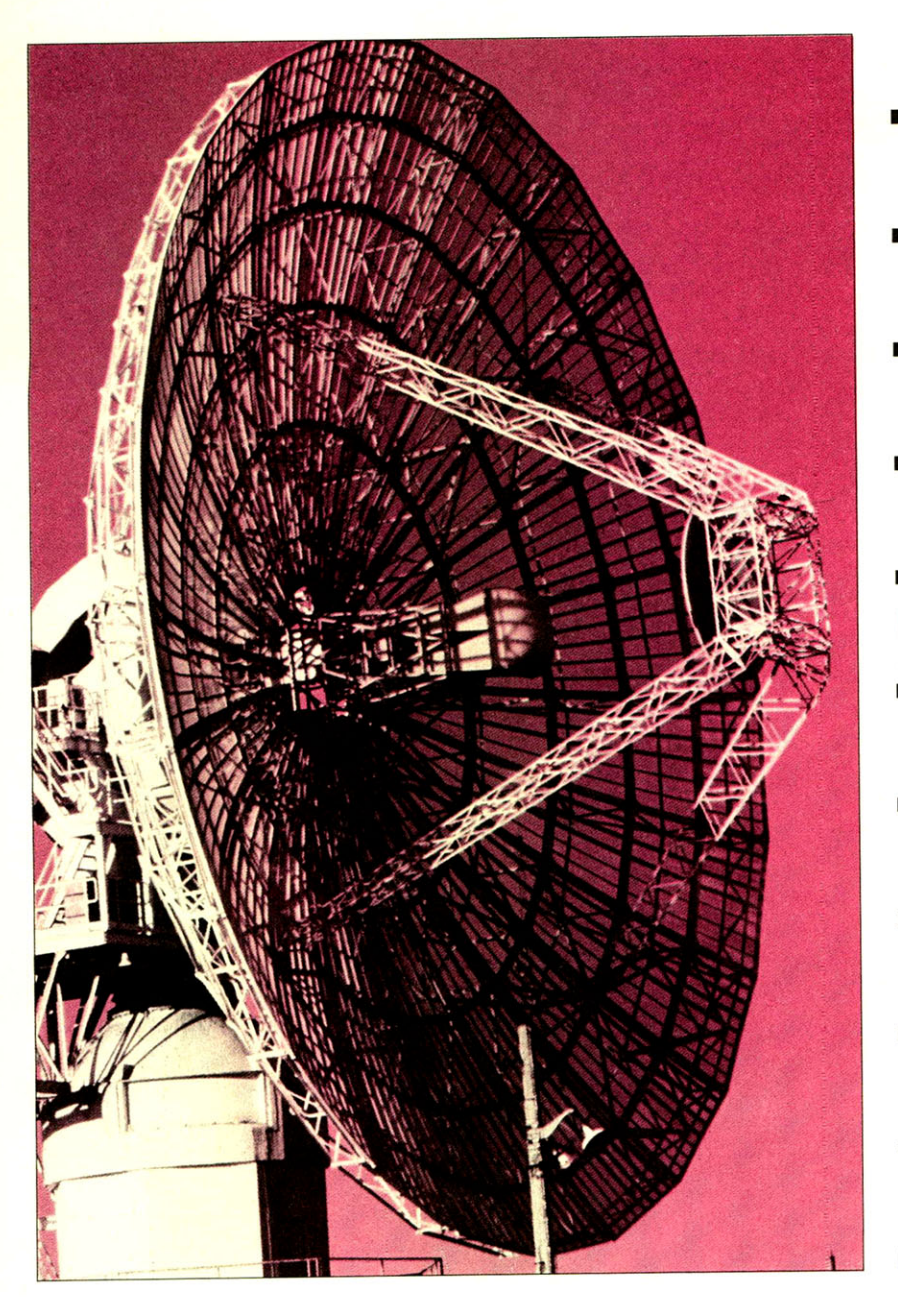
Bei den Kommunikationsdiensten soll nur darauf hingewiesen werden, daß Microcomputer sowohl - wenn auch mit unterschiedlichem finanziellen Aufwand - mit Telex als auch mit Teletex verknüpft werden können, wobei Teletex schon aus finanziellen Gründen der Vorzug zu geben ist. Telefax kommt im juristischen Bereich allenfalls dort in Betracht, wo nicht nur Texte, sondern auch Zeichnungen schnell übertragen werden müssen. Die Einsatzmöglichkeiten für Bildschirmtext im Anwaltsbereich sind umstritten und spielen zumindest gegenwärtig noch keine bedeutende Rolle (16). Als ein Anwendungsbeispiel sei die Entscheidungssammlung des Alfred Metzner Verlages zur Schmerzensgeldrechtsprechung genannt (17).

Eine zunehmende Bedeutung erlangt dagegen das Angebot an externen Datenbanken für Juristen (18). An erster Stelle ist hier JURIS zu nennen (19). In JURIS sind zur Zeit über 560.000 Dokumente gespeichert. Die Rechtsprechungsdatenbank mit über 230.000 Dokumenten dokumentiert die Rechtsprechung des Bundesverfassungsgerichtes und der fünf obersten Bundesgerichte, soweit diese veröffentlicht sind. Ab dem Jahr 1976 ist auch die veröffentlichte Instanzrechtsprechung erfaßt, wobei die arbeitsrechtlich oder sozialrechtlich bedeutsame Rechtsprechung der Instanzgerichte bis 1954 dokumentiert wird. Daneben werden auch unveröffentlichte Entscheidungen nachgewiesen, soweit sie Juris zugänglich gemacht wurden. Abrufbar sind die Leitsätze der Urteile und/oder leitsatzähnliche Kurzfassungen sowie die Fundstellen. Bei Entscheidungen des BFH ist der Volltext gespeichert. Die Literaturdatenbank enthält ebenfalls über 230.000 Nachweise von Veröffentlichungen in 160 Fachzeitschriften seit 1976. Abrufbar sind hier ebenfalls kurze Zusammenfassungen oder Gliederungen sowie die Fundstelle. Die Normendatenbank enthält z.Z. zwei Drittel aller Gesetze und Rechtsverordnungen des Bundes im vollen Wortlaut. Für die Benutzung sind seit Anfang Februar 1987 die Gebiete Staats- und Verfassungsrecht, Rechtspflege, Zivilrecht und Strafrecht sowie Verteidigung zugänglich. Die Verwaltungsvorschriftendatenbank enthält die steuerrechtlichen Verwaltungsvorschriften im vollen Wortlaut, sozialrechtliche Verwaltungsvorschriften im Kurztext mit Fundstelle, jeweils ab 1978. Der Umfang beträgt z.Z. 28.000 Dokumente.

Auf die Frage, ob JURIS als zentrales Volltextdokumentationssystem in der Lage ist, die teilweise hochgesteckten Erwartungen zu erfüllen, kann hier nicht näher eingegangen werden. In diesem Zusammenhang geht es zum einen um die Frage, ob der Zugriff auf informationen über Schlagworte überhaupt der juristischen Arbeitsweise entspricht, um Probleme der Rückgewinnung einmal gespeicherter Dokumente auf mathematischer und linguistischer Ebene (20). Zum anderen wird angesichts der technologischen Entwicklung der Speichermedien der Aufbau einer eigenen, dezentralisierten Datenbank möglich. Bislang ist dies - wie bei BGH-DAT oder WEHRDAT - nur durch den Versand von Disketten möglich, die auf Festplatte überspielt werden. Dahinter lauert aber schon die CD-ROM in Wartestellung. Für Atari ST Besitzer ist daher nicht uninteressant, daß Atari einen recht günstigen CD-ROM/CD-Audio Player angekündigt hat.

Weniger bekannt als JURIS ist die Steuerrechtsdatenbank LEXINFORM der DATEV. Unterschätzt wird allgemein REGISTER der Firma Ecodata, über die alle Handels- und Genossenschaftsregistereinträge des Bundesanzeigers seit 1985 abrufbar sind. Als Suchwort kann hier nicht nur der Name eines Unternehmens sondern auch der einer natürlichen Person verwendet werden, so daß wirtschaftliche Verflechtungen und Beteiligungen transparent werden, was nicht nur für den Erfolg von Zwangsvollstreckungsversuchen bedeutsam ist. REGIST-2 enthält Bekanntmachungen, Konkurs- und Musterregistereinträge im Bundesanzeiger seit 1985. Über HOPPENSTEDT GERMANY können Informationen über größere deutsche Unternehmen (mind. 150 Be-

geht weiter...



...UND PLÖTZLICH HAT IHR ATARI ST ODER MEGA ST EINE BEGEGNUNG DER BESONDEREN ART: VORTEX HD PLUS

Festplatten-Systeme von 20 bis 120 MB!

Das hat es bis jetzt noch nicht gegeben: Ein Festplatten-Programm für den ATARI ST bzw. MEGA ST mit Kapazitäten von 20 bis 120 MB formatierte Speicherkapazität!

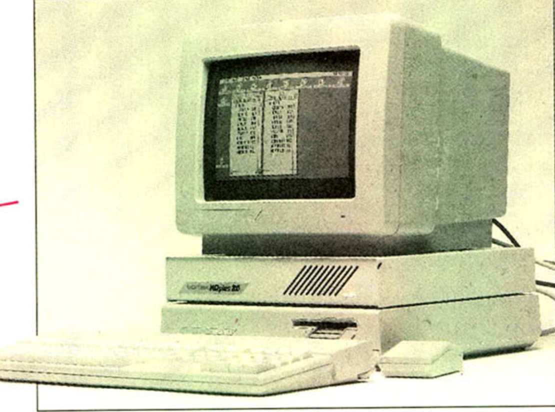
Weitere Vorteile

- Cache-Memory
- Auto-Parker
- bis zu 16 Partitionen
- Disketten-Backup-Programm
- bootfähig

Natürlich im ATARI-Design und in vortex-Qualität. Komplett anschlußfertig mit System-Diskette, Buskabel und deutschem Handbuch.

Holen Sie sich die kompletten Informationen. Sofort!







...UND PLÖTZLICH LEISTET IHR COMPUTER MEHR

I·N·F·O·S·C·H·E·C·K

Senden Sie mir umgehend alle Informationen über Ihr HDplus-Programm und nennen Sie mir den nächstgelegenen vortex-Vertragshändler.

____

____

vortex Computersysteme GmbH
Falterstraße 51–53 · 7101 Flein · Telefon (07131) 52061

schäftigte oder mind. 500.000 DM Grundkapital oder mind. 5.000.000 DM Umsatz) abgerufen werden. Im deutschsprachigen Raum sei noch auf PATDPA (Patentrechtsdatenbank des deutschen Patentamtes mit Rechtsstandsinformationen), PATOS (Patentoffenlegungsschriften des Bertelsmann Informationsservices), GENIOS (Wirtschaftsdatenbank), ASYLDOC (Literatur und Rechtsprechung im Bereich des Asylrechts seit 1983) und CELEX (Recht der europäischen Gemeinschaft) hingewiesen (21). Schließlich gibt es einen Datenbankservice, der Schriftsatzentwürfe für Rechtsanwälte und auch Abwehr-Schriftsätze anbietet (22).

Der Zugriff auf diese externen Datenbanken kann durch einen Mailboxanschluß finanziell attraktiver sein, weil dann ein (gebühren-pflichtiger) Direktanschluß (allein bei JURIS 300,-DM pro Monat) entfällt. Dafür entstehen Kosten für den Mailboxanschluß und natürlich für den Zugriff selbst. Mailboxen ermöglichen darüberhinaus Textkommunikation und Nachrichtenaustausch über sog. "elektronische Briefkästen". Diese Art der Kommunikation steckt gegenwärtig - zumindest im juristischen Bereich - noch in den Kinderschuhen. Sie eröffnet zukünftig jedoch die Möglichkeit eines raschen Informationsaustausches im Verkehr mit Mandanten, Korrespondenzanwälten und Gerichten sowie in der Kautelarjurisprudenz durch Computerkonferenzen und - vertragsverhandlungen. Darüberhinaus können bestimmte Teilnehmergruppen, Standesorganisationen etc. an sog. "Schwarzen Brettern" allgemein zugängliche Informationen veröffentlichen. Als Anbieter sind im juristischen Bereich die Jur-BOX der Deutschen Mailbox und ALexIS der Hans-Soldan-Stiftung (unter Beteiligung der Bundesrechtsanwaltskammer) interessant (23). Beide Mailboxen bieten noch weitere Leistungen wie beispielsweise Übersetzungen durch ein Tele-Büro. ALexIS bietet darüberhinaus Recherchen in Datenbanken, elektronische Bestellungen bei der Hans-Soldan-Stiftung (24), elektronische Buchungen bei dem Anwaltsinstitut der Bundesrechtsanwaltskammer und Telesatz. Welche faszinierenden weiteren Entwicklungsmöglichkeiten im Mailbox-System stecken, zeigt ein Blick auf das ABA/net, das Mailbox-System der American Bar Association (25).

5. Juristische Ausbildung

Die Vermittlung sowohl von "Beherrschungswissen" als auch von "Handhabungswissen" im Hinblick auf elektronische Datenverarbeitung zählt in der gegenwärtigen Juristenausbildung weder zu den Pflicht- noch zu den Wahlfächern. Dieses Wissen ist daher auch nicht "prüfungsrelevant" (26). Über den Pflichtenkanon der juristischen Ausbildung hinausgehende zusätzliche Lehrveranstaltungen während der universitären oder der Referendarausbildung haben bislang noch Seltenheitswert.

Im universitären Bereich ist eine langsame Steigerung des Ausbildungsangebotes zu verzeichnen. Im Sommersemester 1987 bestand an 15 der 31 juristischen Fakultäten die Möglichkeit, einschlägige Veranstaltungen zu besuchen (27). Von den insgesamt 24 Veranstaltungen wurden 8 in Vorlesungs- und 12 in Seminarform abgehalten, bei den restlichen Veranstaltungen handelte es sich um eine Arbeitsgemeinschaft, ein Kolloquium, eine Übung und einen Programmierkurs. Der Schwerpunkt der Veranstaltungen lag bei Einführungen in die Rechtsinformatik bzw. Datenverarbeitung (16 Veranstaltungen). Daneben wurden aber auch Veranstaltungen über PC-Anwendungen im Recht (München), Textbe- und -verarbeitung für Juristen (Konstanz), wirtschaftliches Rechnungswesen und EDV, strafrechtliche und strafprozessuale Probleme der EDV (beide München), Datenschutzrecht (Bonn und Hannover) und über juristische Expertensysteme (Hannover und Konstanz) angeboten.

Bei diesen Veranstaltungen handelt es sich - soweit nicht auf andere universitäre Einrichtungen zurückgegriffen werden kann - überwiegend um "Trockenkurse". Über ein eigenes PC-Labor verfügt gegenwärtig erst eine einzige juristische Fakultät (Tübingen). An weiteren vier juristischen Fakultäten (Göttingen, Konstanz, Marburg, Münster) ist die Einrichtung derartiger Labors im Gange.

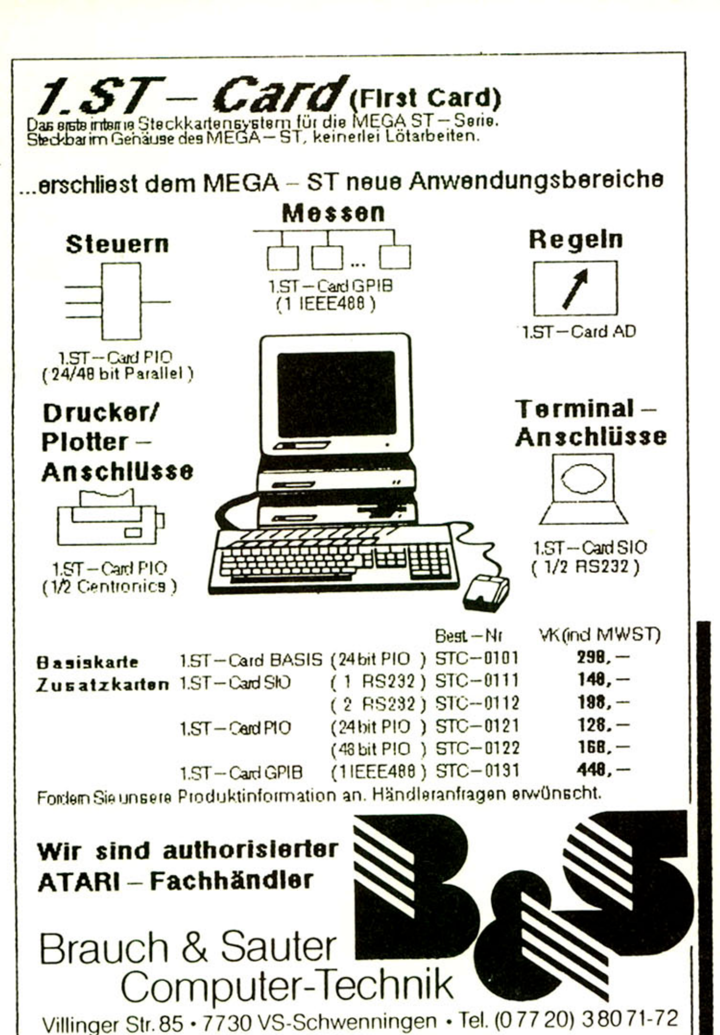
Eine EDV-Ausbildung für Referendare ist dagegen nach wie vor nur vereinzelt anzutreffen. Eine Ausnahme stellt insbesondere die Hochschule für Verwaltungswissenschaften in Speyer dar. Dort werden seit über zehn Jahren Vorlesungen, Seminare und praktische Übungen für Rechtsreferendare angeboten. In der übrigen Referendarausbildung liegt Hessen "vorn". Im Hessischen Justizministerium wurde 1985 eine Arbeitsgruppe "EDV in der Juristenausbildung" eingerichtet. Nach einem Pilotprojekt wird beim Landgericht Frankfurt eine freiwillige praktische Studienzeit "Informationstechnik für Juristen" angeboten. Ähnliche Versuche in Form von freiwilligen Arbeitsgemeinschaften gab es beim Hanseatischen Oberlandesgericht in Hamburg und am OLG München. Im übrigen erfährt der Referendar im Vorbereitungsdienst selbst dann kaum etwas über die elektronische Datenverarbeitung, wenn diese schon in der Rechtsanwaltskanzlei seiner Ausbildungsstation eingeführt wurde. Denn in der Regel hat der Referendar mit der Routinetätigkeit der Dezernatsarbeit des Richters oder des Rechtsanwalts wenig zu tun. Die einseitige Ausrichtung der Referendarausbildung auf das Schreiben von Gutachten und Entwürfen rächt sich jedoch spätestens beim Beginn der selbständigen Tätigkeit nach Abschluß der juristischen Ausbildung.

Für Juristen, die nach dem Abschluß ihrer Ausbildung keinen oder nicht sofort einen Arbeitsplatz finden, gibt es eine Reihe von Möglichkeiten, sich auf dem Gebiet der elektronischen Datenverarbeitung zusätzlich zu qualifizieren. Neben dem Aufbaustudiengang "Grundzüge der Datenverarbeitung" (Marburg) bzw. dem Zusatzstudiengang "Informatik für Geisteswissenschaftler (Fachrichtung Rechtswissenschaft)" (Bochum) (28) ist auf praxisorientierte Schulungsmaßnahmen hinzuweisen, die von den Arbeitsämtern in Zusammenarbeit mit privaten Weiterbildungseinrichtungen (gelegentlich Ableger von Computerfirmen) veranstaltet werden (29).

Zusammenfassend kann festgestellt werden, daß an den juristischen Fakultäten die Vermittlung von Wissen über elektronische Datenverarbeitung nur eine marginale Bedeutung hat. Dort, wo Veranstaltungen angeboten werden, dominieren Einführungen in Rechtsinformatik oder Datenverarbeitung. Die Vermittlung von Fertigkeiten, also Handhabungswissen, ist schon aufgrund der sachlichen Ausstattung nur an wenigen juristischen Fakultäten möglich. Das Angebot an Lehrveranstaltungen scheint stark von dem Engagement einzelner Professoren abzuhängen.

Vor diesem Hintergrund kann jedem Jurastudenten nur dringend geraten werden, die spärlichen Angebote im Rahmen seiner Ausbildung wahrzunehmen, und sich darüberhinaus auf eigene Faust mit der elektronischen Datenverarbeitung vertraut zu machen. Im Klartext bedeutet dies die Anschaffung eines Computers, wobei der

geht weiter...



			Best – Nr	VK (incl MWST)
Basiskarte	1.ST – Card BASIS	(24 bit PIO)	STC – 0101	298,–
Zusatzkarten	1.ST – Card SIO	(1 RS232)	STC – 0111	148,–
		(2 RS232)	STC – 0112	198,–
	1.ST – Card PIO	(24 bit PIO)	STC – 0121	128,–
		(48 bit PIO)	STC – 0122	168,–
	1.ST – Card GPIB	(1 IEEE488)	STC – 0131	448,–

COPY ST V1.2

DAS SUPER KOPIERPROGRAMM

- ★ V1.2 - STARK VERBESSERT
- ★ MACHT DORT WEITER WO DIE ANDEREN AUFHÖREN
- ★ KOPIERT FAST ALLE ST-DISKETTEN
- ★ BESITZT EINSTELLUNG FÜR START- UND ENDTRACK
- ★ IST VOLL GEM-UNTERSTÜTZT; DADURCH SEHR EINFACH IN DER HANDHABUNG
- ★ HAT EINE AUTOMATISCHE FEHLERERKENNUNG; DADURCH KEINE PARAMETERANGABE NOTWENDIG
- ★ EIGENE FORMATIERROUTINE GIBT BIS ZU 230 KB BZW. 130 KB MEHR DISKETTENKAPAZITÄT
- ★ HAT EIN UPDATESERVICE
- ★ FÜR EIN UND ZWEI LAUFWERKE, EIN- U. DOPPELSEITIG
- ★ DAS BESTE AUF DEM DATENSICHERUNGSGEBIET

PREIS NUR ★ 98,- DM ★

DISKTOOL ST

- ★ DAS PROFESSIONELLE DISKTOOL
- ★ VOLL GEM-UNTERSTÜTZT
- ★ EINLESEN UND EDITIEREN EINZELNER SPUREN UND ADRESSFELDER
- ★ EINLESEN UND EDITIEREN VON SEKTOREN
- ★ VERGLEICHEN VON DISKETTEN UND EINZELNER FILES
- ★ SUCHFUNKTION
- ★ EINLESEN, EDITIEREN, ABSPEICHERN EINZELNER FILES
- ★ USW., ...

PREIS NUR ★ 89,- DM ★

E U R O S Y S T E M S
HOLLAND

FILIALE FÜR DEUTSCHLAND:
BREDENBACHSTRASSE 129, 4240 EMMERICH
TEL. TÄGLICH 14-18 UHR 02822/52151

SOFT- UND HARDWARE-NEUENTWICKLUNGEN FÜR ATARI ST UND AMIGA GESUCHT. BESTELL.: BEI VORKASSE: 48-STUNDEN-SERVICE (WENN LAGERND), KOSTEN DM 4,- NACHNAHME: KOSTEN DM 8,-. AUSLAND: NUR VORKASSE, EUROCH., POSTANW.

DISTRIBUTOR FÜR DIE SCHWEIZ:
NAUER DESIGN, DORFSTRASSE 28, CH-4612 WANGEN, TEL. 06232/2858

vortex VERSAND

DER

COMPUTER-PERIPHERIE · SOFTWARE

KA

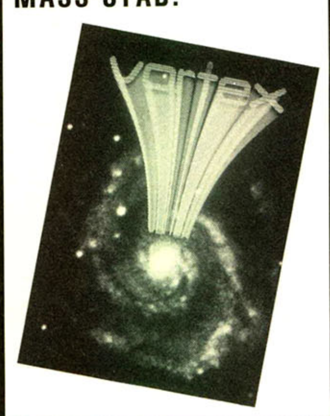
REINIGUNG

TA

UND PFLEGE

LOG

DER VORTEX-VERSAND IST EIN TOCHTERUNTERNEHMEN VON VORTEX, DER BEKANNTEN MARKE FÜR COMPUTER-PERIPHERIE. QUALITÄT IST UNSER MASS-STAB.



UNENTBEHRLICH FÜR ALLE ATARI ST-BESITZER.

KATALOG-ANFORDERUNG

Bitte senden Sie mir postwendend Ihren Versand-Katalog für Atari ST. Die Schutzgebühr in Höhe von DM 3,– lege ich in Briefmarken bei.

Name ____________________

Straße ____________________

Ort ____________________

Telefon ____________________

vortex-Versand
Falterstraße
7101 Flein

Atari ST sich aufgrund seines günstigen Preis-/Leistungsverhältnisses anbietet und auch für Studenten erschwinglich ist. Erheblich stärker zu Buche schlägt die Anschaffung eines Druckers, der gut lesbare Ausdrucke produziert. Hier besteht an manchen Universitäten die Möglichkeit, Ausdrucke über das Hochschulrechenzentrum, die Fakultäten o.ä. zu erstellen. Bei der Nutzung eines eigenen Computers kann es zunächst nur darum gehen, sich elementares "Handhabungswissen" im Hinblick auf spätere berufliche Anwendungen zu erwerben. Dies betrifft einmal Kenntnisse der Textverarbeitung, die sich beim Schreiben von Hausarbeiten, Referaten und Examensarbeiten erwerben lassen. Hier bietet sich als Textverarbeitungsprogramm "Wordplus" an, das sehr schnell erlernbar ist und außer im Bereich der Fußnotenverwaltung wenig Wünsche offenläßt. Zum zweiten wird jeder Student, zumindest im vorgerückten Semester, seine eigene Datenbank zur Verwaltung relevanter Entscheidungen und Literatur anlegen. Insoweit wird auf die obigen Ausführungen verwiesen. Als dritter Einsatzbereich kommt der Einsatz spezieller Lernsoftware in Betracht, für Jurastudenten also die bereits angesprochene "JURA Lern- und Definitionskartei" (30).
Nach diesem Überblick über die Anwendungsgebiete des Computers im juristischen Bereich wird es in den Folgebeiträgen darum gehen, wie weit diese Aufgaben vom Atari ST im Zusammenwirken mit geeigneten Anwendungsprogrammen übernommen werden können.

Joachim Kleveman,
Wiss. Mittarbeiter,
Universität Bielefeld

Verzeichnis der Fußnoten:

(1) Der aktuellste und fundierteste Überblick über Rechtsanwaltsprogramme findet sich in dem "EDV-Check-Buch für Rechtsanwälte 1986/87" von Benno Heussen, München 1986. Siehe außerdem Sayeed Klewitz "Bürocomputer in der Anwaltskanzlei" in: AnwBl 1984(4), 179-183. Dort finden sich ausführliche Literaturnachweise über Erfahrungsberichte mit einzelnen Softwareprogrammen. Bei deren Lektüre sollte man bedenken, daß kaum ein Verfasser eine Fehlinvestition öffentlich eingestehen würde. Abgesehen davon ist der Umfang der interessenmäßigen Identität zwischen Verfasser eines Berichts und dem Anbieter eines Programmes von außen kaum zu beurteilen.

(2) Winterhuder Weg 92, 2000 Hamburg 76. Ein Kostenprogramm zur Unter-stützung der Erstellung von Kostenfestsetzungsgesuchen und Rechnungen ist in Vorbereitung. Desweiteren wird an die Einbeziehung der ADIMENS ST Datenbank in die Rechtsanwaltsprogramme gedacht. Unter dieser Adresse kann auch ein "Anwalts - Info: ATARI ST-Computer im Anwaltsbüro" bezogen werden.

(3) Vertrieb: Werner Forkel, Kriegsstr. 29, 7500 Karlsruhe 1.

(4) Dazu Benno Keim "Über das Arbeiten mit Textbausteinen" in: AnwBl 1984(1), 1-11 und Helmut Becker "Juristische Arbeit und Textverarbeitung" in: CuR 1986(9-11), 607-610, 685-690 und 767-
772. Siehe in diesem Zusammenhang auch die "Diktat- und Arbeitsbücher" im C.H.Beck Verlag.

(5) Die Möglichkeiten des EDV-Einsatzes in der Kautelarjurisprudenz sind beispielhaft beschrieben bei Carsten Thomas Ebenroth und Helmut Becker "EDV-gestützte Gestaltung internationaler Verträge" in: CuR 1986(8), 504-510.

(6) Hier ist das Programm "Lidos" positiv zu erwähnen, dazu Joachim Brunold "Literaturdokumentation mit dem PC" in: BuchMarkt 1987(4), 182-191. LIDOS ist neuerdings auch in einer kostengünstigen Version für den Atari ST erhältlich (LIDOS ST, ca. 900,- bei: EXpress Edition, Ritterstr. 60b, 1000 Berlin 61, 25v.H. Hochschulermäßigung).

(7) Zu weiteren Schwierigkeiten siehe Peter Waltl "Der Aufbau einer kanzleiinternen Datenbank - Erfahrungen und Perspektiven" in: CuR 1987(8), 550-553.

(8) Die "Neue Juristische Wochenschrift" ist die meistgelesene juristische Zeitschrift. In der NJW-Leitsatzkartei wird die aktuelle Rechtsprechung und Literatur auf 3,5 x 5,5 cm großen Karteikarten dokumentiert. In jedem Heft ca. 210 Nachweise führen bei wöchentlicher Erscheinungsweise zu über 10.000 Karteikarten pro Jahr.

(9) "BGH-DAT" wird vorläufig nur für die Atari ST-Computer angeboten. Als Datenbank dient das auf diesem Betriebssystem führende ADIMENS ST-Programm. Für deren "EXEC"-Modul ist eine Lizenzgebühr von 100,- DM zu entrichten. Der Jahrgang 1986 kostet 599,- DM, ab 1.11.1987 750,- DM, jede vierteljährliche Ergänzungsdiskette 150,- DM. Eine Demo-Version für IBM-kompatible Personalcomputer soll demnächst vorliegen. Siehe hierzu die Berichte in Atari Aktuell, Ausgabe März 1987, 10 und von Maximilian Herberger in Infojur 1987(10), 359-360.

(10) Auch WEHRDAT ist eine ADIMENS ST Applikation. Mit ADIMENS ST (Vers. 2.1) kostet es 598,- DM, ohne ADIMENS ST 398,- DM (sowohl für ADIMENS ST Vers. 1.6 oder 2.1. Eine Demo-Version kostet 50,- DM, der Preis wird beim Kauf angerechnet. Vertrieb: Werner Forkel, Kriegsstr. 29, 7500 Karlsruhe 1. Siehe dazu auch den Bericht in: Atari Aktuell, Ausgabe September 1987, 9.



Weiterhin ist auf die Datenbank ASYLDOC der Zentralstelle der freien Wohlfahrtspflege für Flüchtlinge e.V. hinzuweisen, die ebenfalls in Diskettenform, jedoch nur für MS-DOS-kompatible Hardware bezogen werden kann. Als Datenbankprogramm dient hier "PC-File", ausführlich Bernd Martens-Parrée "Das Informationskarteikartensystem der ZDWF" in: infojur 1986(10), 405-413 m.w.N.

(11) Siehe zum Vorstehenden auch Günter Ernesti "Der Einzug der elektronischen Datenverarbeitung in den Bereich der Justiz" in: DRiZ 1987(4), 129-138.

(12) Armin Nack "Computereinsatz in Wirtschaftsstrafverfahren" in: DRiZ 1985(11), 425-428; ders. "Judex computat" in: DRiZ 1986(11), 406-415, unter der Überschrift "Richter am Computer. Der Computer als Hilfsmittel des Richters: Aktueller Stand und Perspektiven" in: Computergestützte Expertensysteme, Ulrich Erdmann; Herbert Fiedler, Fritjof Haft, Roland Traunmüller (Hrsg.), Tübingen 1986191-220; Lutz van Raden "Arbeitsplatzcomputer in der Justiz" in: Infojur 1987(6u.7), 217-221 u. 263-267.

(13) Allgemein zu VIP Professional siehe nur Klaus Heuer "VIP Professional im GEM Gewand" in: ST-Computer 1987(3), 78-82.

(14) Siehe nur Peter Friederici "Der Einsatz des PC im Versorgungsausgleich" (Teil 1-3) in: Infojur 1986(1-3), 39-42, 90-91 u. 126-129.

(15) Axel Bauer "Was können moderne elektronische Medien für das Anwaltsbüro leisten?" in: BRAK-Mitt. 1985(4), 192-197 sowie Eva Wolf "Neue Kommunikationstechniken im Anwaltsbüro - Teletex, Telefax, Telebox" in: AnwBl. 1986(3), 126-130. Zu den rechtlichen Voraus-setzungen des elektronischen Dokumentenaustausches im Verhältnis Anwaltschaft/Justiz, Anwaltschaft/Grundbuchamt und innerhalb der Anwaltschaft siehe die sog. "URDOK-Studie". Die wesentlichen Auszüge sind von Ulrich Seidel unter dem Titel "Urkundensichere Dokumentenverarbeitung" veröffentlicht in: BRAK-Mitt. 1987(1), 11-15. Hinsichtlich des Betreibens von Btx am Atari ST siehe die Kurzberichte in ST-Computer 1987(10), 11 und 1987(12), 10.

geht weiter...

(16) Siehe dazu Ottheim Kääb "Btx und die Bedeutung für die Rechtsanwaltschaft" in: BRAK-Mitt. 1984(2), 46-47; Jens-Peter Lachmann und Jochen Schneider "Wie ist Bildschirmtext (Btx) für den Anwalt als Hilfsmittel zu nutzen?" in: AnwBl. 1985(8u.9), 460-472; Maximilian Herberger "BTX - bereits ein juristisches Fachinformationssystem?" in: Infojur 1986(1), 45-49; Andreas Herberger "Bildschirmtext - bereits ein juristisches Fachinformationssystem? Teil 2: Antworten auf eine redaktionelle Umfrage?" in: Infojur 1986(3), 129-132; Rolf Lichtner "Btx als Medium der Anwaltschaft" in: CuR 1987(2), 133-136.

(17) Dazu Andreas Herberger "Schmerzensgeldrechtsprechung Online" in: Infojur 1986(2), 97-99.

(18) Siehe nur den Übersichtsartikel von Johann Tilling und Ralf Abel "Juristische Datenbanken - Die Zukunft hat begonnen" in: AnwBl. 1986(3), 130-134.

(19) Die Literatur zu JURIS ist mittlerweile fast nicht mehr zu überblicken, daher sei lediglich auf das Informationsmaterial verwiesen, das bei der JURIS GmbH Juristisches Informationssystem für die Bundesrepublik Deutschland, Gutenbergstr. 23, 6600 Saarbrücken angefordert werden kann. Ein konzentrierter Überblick findet sich bei Winfried Schreiber "JURIS - Das Rechtsinformationssystem für die Bundesrepublik Deutschland - Informationsangebot, Arbeitsweise, Zugang -" in: AnwBl. 1986(1), 29-30. Zur Dokumentationsdichte der Entscheidungen der obersten Bundesgerichte



im Informationssystem JURIS siehe den gleichlautenden Artikel von Jörg Berkemann und Karin Siebert in: CuR 1987(6), 385-393.

(20) Siehe hierzu nur Fritjof Haft "Computer können die Rechtsanwendung verändern - nicht nur im Steuerrecht" in: DSWR 1987(4), 73-79(75); David Blair und M.E. Maron "An Evaluation of Retrieval Effectiveness for a Full-Text Document Retrieval System", 28 COM. A.C.M. 289 (1985) (publication of the Association for Computing Machinery), zitiert nach Robert C. Berring "Volltext-Datenbanken und juristische Informationssuche: Mit dem Rücken zur Zukunft" in: Infojur 1987(1-3), 5-11, 70-75 und 115-123(73-75); zuerst in: High Technology Law Journal 1986(1), 27-60.

(21) Zu LEXINFORM siehe Wilhelm H. Wacker und Kurt-Dieter Koschmieder "Steuerrechtsdatenbank LEXinform" in: DSWR 1984(11), 245-248; Joachim Conradi "Die Steuerrechtsdatenbank LEXinform" in: Rechtsinformatik in den achtziger Jahren, Hrsg. Hermann Seegers und Fritjof Haft, München 1984, 31-42, Gerhard Igl "Die Steuerrechts-Datenbank LEXinform" in: CuR 1986(1), 54-55; zu REGISTER die redaktionelle Mitteilung "Elektronisches Handelsregister" in: BRAK-Mitt. 1985(4), 209; zur PATDPA siehe Alfred Wittmann PATDPA - eine deutsche Patentdatenbank zur Ermittlung des Standes der Technik und der Patentlage" in: CuR 1987(2), 137-141; zu PATOS siehe die redaktionelle Kurzmitteilung in: CuR 198785), 332; zu GENIOS siehe Andreas Herberger "Die GENIOS-Wirtschaftsdatenbank (Teil I)" in: Infojur 1986(5), 209-213; zu ASYLDOC siehe Ilse Bueren "ASYLDOC - Eine Datenbank (Teil 1)" in: Infojur 1986(2), 85-86 und Bernd Martens-Parrée "Die Datenbank ASYLDOC (Teil 2 und 3)" in: Infojur 1986(3u.4), 117-122 u.172-174; Zu europäischen und Datenbanken in den USA siehe Jürgen Farrenkopf "Rechtsinformationen durch Online-Datenbanken in Europa - Ein Überblick - " in: DSWR 1986(5), 107- 111 m.w.N.; o.V. "Die Euronet-Datenbanken mit juristischer Thematik" in: Infojur 1986(10), 413-416 und Andreas Herberger "Der Zugriff auf internationale Telekommunikationsnetze" in: infojur 1986(7-8), 332-334.

geht weiter...

Mini Clip

Der ATARI ST in der Praxis

Unter diesem Motto möchten wir eine Reihe von praktischen Anwendungen auf dem ATARI ST vorstellen. Den Anfang machte jetzt der Artikel ´JuriSTische Anwendungen´. Damit dies nicht der letzte seiner Art bleibt, möchten wir hiermit Sie, liebe Leser, dazu aufrufen, uns Ihre Erfahrungen und Anwendungen mit dem ATARI ST im alltäglichen Leben mitzuteilen.

Vielleicht machen Sie Ihre Buchführung, verwalten Ihre Kunden oder steuern Ihre Meßgeräte mit dem ST. Alles, was Sie für einen Leserkreis, wie den der ST-Computer, interessant halten, können Sie uns mitteilen. Im Zweifelsfalle rufen Sie einfach die Redaktion unter der aufgeführten Adresse an.

Veröffentlichte Erfahrungsberichte werden natürlich angemessen honoriert.

Bitte beachten Sie bei der Einsendung:

Schicken Sie den Bericht und eventuelle Hardcopies auf Papier und Diskette. Die Diskette wird zurückerstattet.

Einsendungen direkt an die Redaktion:

Merlin-Computer GmbH
ST-Computer Redaktion
´Anwendungen´

Industriestr. 26
D-6236 Eschborn
Tel.: 06196/481811

(22) Nach einer Anzeige der COMPIURA DATENBANKSERVICE GMBH in: NJW 1986(47), XVII.

(23) Zu ALexIS siehe Axel Bauer "ALEXIS Das Anwalts-Rechts-Informationssystem" in: CuR 1986(6), 360-364; Rolf Lichtner "ALEXIS - die Anwaltsmailbox. Kommunikations- und Informationssystem für die Anwaltschaft" in: BRAK-Mitt. 1986(2), 70-74 und Wolfgang Andreae "Was kostet ALexIS?" in CuR 1987(1), 71-72. Nach einem Bericht von Bauer auf der Infobase '87 hat ALexIS z.Z. ca. 150 Mitglieder, Jörg-Martin Schulze "Infobase '87: Juristensymposium" in: Infojur 1987(6), 150-252(252). Informationen über die Jur-Box sind unter der Adresse: Deutsche Mailbox GmbH, Postfach 500324, 2000 Hamburg 50 erhältlich; dazu auch der redaktionelle Kurzbericht in: CuR 1987(6), 395.

(24) In Diskettenform wird dieser Service schon von der Juristischen Fachbuchhandlung am Landgericht GmbH, Zweigertstr. 31, 4300 Essen 1 unter dem Namen "JUDID" angeboten.

(25) Dazu der informative Beitrag von Maximilian Herberger "ABA/net - Das Netzwerk der "American Bar Association" in: Infojur 1986(5), 221-228. Als Beispiel eines allgemein ausgerichteten Mailbox-Kommunikationsnetzes siehe ders. "The CompuServe Information Service - Ein elektronisches Informations- und Telekommunikationsnetz" in: Infojur 1986(3), 133-136 m.w.N. und ders. ""LAWSIG" (Law Special Interest Group) - Die juristische Arbeitsgruppe in CompuServe" in: Infojur 1986(4), 179-182.

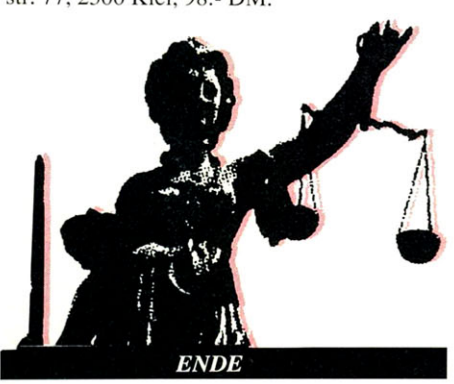
(26) So Wolfgang Heinz "Iudex non calculat - oder: Weshalb soll und wie kann den Juristen die Angst vor dem Computer genommen werden?" in: Infojur 1987(6), 242-247(247) (unter dem Titel "Lesen und Schreiben müssen durch Computer-Kompetenz ergänzt werden" auch in: Juristische Nachrichten 1987(II), 13).

(27) Ausführlich Heinz a.a.O.

(28) Vgl. Burkhard Piel "Zusatzdiplom "Informatik für Geisteswissenschaftler" an der Ruhr-Universität Bochum" in: Infojur 1987(4), 163-165.

(29) Beispielsweise Maximilian Herberger (mh) "Neue Zusatzqualifikation für Juristen: Wirtschaftsjurist - EDV" in: Infojur 1986(7/8), 336.

(30) Vertrieb: STUDIOSUS OHG, Olshausenstr. 77, 2300 Kiel, 98.- DM.



CAD Project

ein objektorientiertes Zeichenprogramm

Spätestens seit Einführung des Mega-ST ist dieser Rechnertyp auch für anspruchsvolle Problemlösungen geeignet. Vor allem im professionellen Bereich machen sich die Vorteile der Speicherriesen bemerkbar. So buhlen denn auch verschiedene Textverarbeitungen, Buchhaltungsprogramme und Datenbanken um die Gunst des Käufers. Auf dem Gebiet der CAD-Programme jedoch geschieht wenig. Bereits im September 87 angekündigt, aber erst um die Jahreswende ausgeliefert, liegt uns jetzt der neueste Vertreter dieser Klasse zum Test vor: CAD project von Schwarzstein Software.

Eins vorweg: Wollte man dieses Produkt aus München wirklich ausführlich beschreiben, so hätte man sicher die ganze ST-Computer zu füllen. Daher sollen im folgenden auch nur die herausragenden Punkte von CAD Project Erwähnung finden.

Wir testeten die Version 1.1 von CAD project. Die Auslieferung einer von Fehlern bereinigten Version und eines völlig überarbeiteten Programms ist von den Autoren bereits angekündigt worden. Wahrscheinlich werden diese bei Erscheinen der aktuellen Ausgabe der ST-Computer schon erhältlich sein.

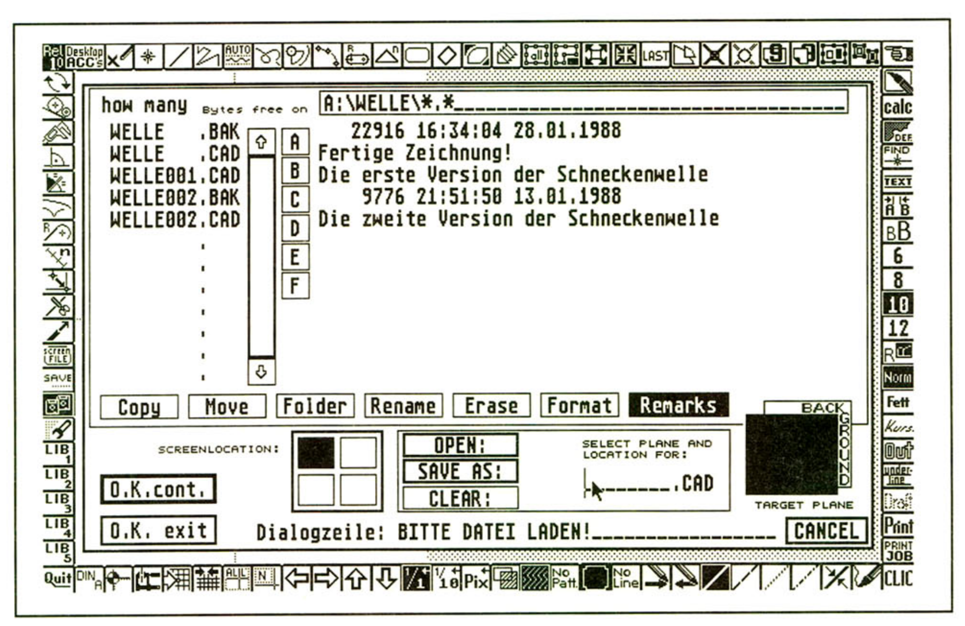


Bild 1: Die Dateiauswahlbox von CAD Project

Zum Lieferumfang gehört neben einem noch recht mageren Handbuch in Ringbuchform (Schwarzstein kündigt auch hier umfangreiche Verbesserungen an) die Programm-Diskette nebst einer 'Keydisk', die kopiergeschützt ist. CAD project selbst ist kopierbar, allerdings muß nach dem Programmstart immer kurz die Keydisk ins Laufwerk A gelegt werden. Als Hardware benötigt der Benutzer einen ST mit mindestens 1 Megabyte Ram und Monochrom-Monitor. 1 Megabyte freier Speicher ist notwendig, da CAD project mit über 500 kByte recht lang geraten ist.

Momentan sind zwei Ausführungen des Programms erhältlich. Die Vollversion für 798 DM enthält Plottertreiber, Berechnungsfunktionen sowie automatische Bemaßung, um die die Normalversion für 298 DM 'erleichtert' wurde.

Das 100 Gänge-Menü

Nach dem Doppelklick auf 'Project.Prg' vergeht zunächst einige Zeit des Ladens. Schließlich bedient sich das Programm noch einiger Dateien wie Zeichensätze, Parameter etc., die mitgeladen werden müssen. Nach der Prozedur mit der Keydisk ist es endlich soweit: das erschreckte Auge des Benutzers erblickt einen Menürahmen mit sage und schreibe 100 Icons.

Doch keine Angst, so unübersichtlich und kompliziert wie diese Art der Benutzerführung zunächst scheint, so schlüssig und einfach wird sie bereits nach kurzer Einarbeitungszeit. Die Funktionen sind sauber gruppiert und die winzigen Symbole durchaus aussagekräftig gewählt. Bei den meisten Funktionen gilt, daß die rechte Maustaste eine Eingabe der Koordinaten in einer Dialogbox ermöglicht, während nach Anwahl mit der linken die Maus als Eingabemedium fungiert.

EasyDraw läßt grüßen Beim ersten Zeichnen bemerkt man, daß bei der Konzeption von CAD project EasyDraw und andere objektorientierte Graphikprogramme Pate standen. Jedes gezeichnete Objekt wird

durch Anklicken mit der Maus von einem Edierrahmen (auch ‘Sizebox’ genannt) umgeben. Dieser rechteckige Rahmen aus dünnen Linien besitzt an seinen Ecken und in der Mitte der Seiten kleine Felder. Werden diese Felder mit der Maus ‘festgehalten’, so läßt sich das betreffende Element der Zeichnung waagerecht, senkrecht oder in Richtung beider Achsen gleichzeitig dehnen und stauchen. Ferner kann das Objekt durch Klick ins Innere des Rahmens verschoben werden.

Soweit die bekannten Funktionen, derer sich auch die meisten Desktop-Publishing-Programme bedienen. Bei CAD project allerdings hat es dieser Rahmen in sich. Wurde ein Objekt ausgewählt, so lassen sich allein mit der Maus nicht weniger als sechs verschiedene Funktionen darauf ausüben. Neben den Achsstreckungen und dem Verschieben sind das: zentrische Streckung (die Proportionen des Objekts bleiben erhalten), Punktspiegelung sowie Achsspiegelung. Außerdem kann durch Druck auf beide Maustasten der Menürahmen ausgeblendet werden. Dadurch ist der Blick auf die gesamte Zeichnung frei.

Aber noch mehr, aus o.g. Programmen bekannte, Funktionen finden sich wieder. Auch hier können mehrere Objekte zu einem zusammengebunden werden. Das bietet sich bei Teilzeichnungen an, die kopiert oder verschoben werden sollen. Selbstverständlich ist es auch möglich, solche Gruppen wieder zu trennen, um z.B. nur ein Detail zu ändern. Objekte können auf Mausklick in den Hinter- oder Vordergrund gebracht werden. Es spielt also die Reihenfolge des Zeichnens bei der Frage, welches Objekt ein anderes überdeckt, keine Rolle.

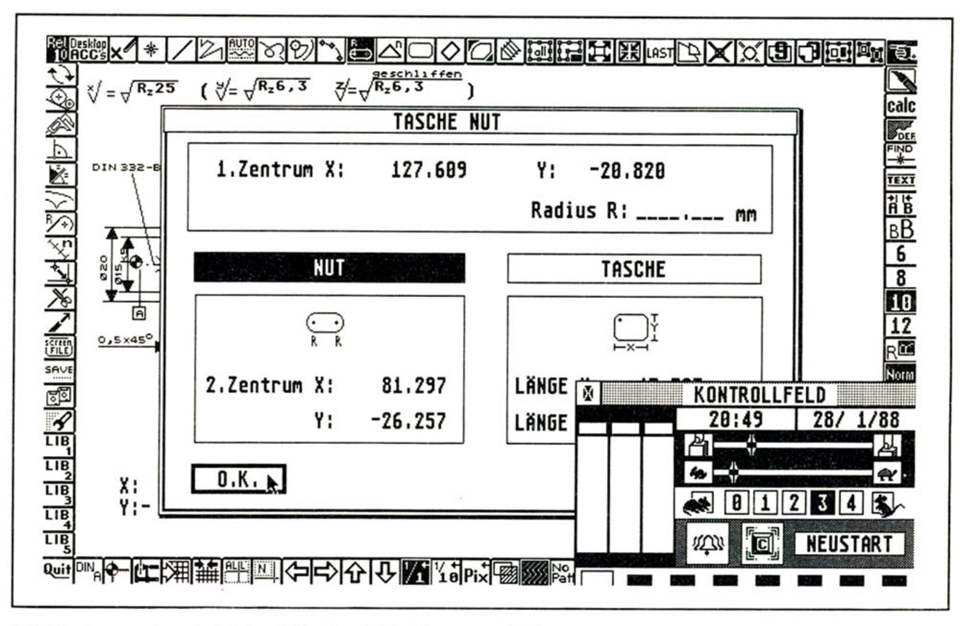


Bild 2: Accessories sind bei geöffneten Dialogboxen verfügbar

Das ‘Einsammeln’ mehrerer Objekte ist bei CAD project einfacher als gewohnt. Wie auf dem Desktop können mit gedrückter Shift-Taste nach und nach Elemente der Zeichnung ausgewählt werden, ohne daß die vorherigen deselektiert werden.

Die Liebe steckt im Detail In CAD project sind eine Reihe von Feinheiten enthalten, die die Arbeit beim Zeichnen erheblich erleichtern. Da ist zunächst die eigene Dateiauswahlbox (siehe Bild1). Mit ihr ist es nicht nur möglich, Dateien zu laden und zu speichern. Neben dem Kopieren, Umbenennen und Löschen lassen sich auch Zeichnungsdateien mit Kommentaren versehen. Wer kennt nicht das Problem, mit den acht Zeichen, die das Betriebssystem nun mal nur zuläßt, eine Datei eindeutig zu benennen.

Während der Arbeit an einer Zeichnung hat man häufig das Problem, durch versehentliches Klicken mit der Maus ein ‘Mini-Objekt’ gezeichnet zu haben. Dieses muß meist umständlich gelöscht werden. In CAD project ist ein Algorithmus enthalten, der solche Fehleingaben prüft und gar nicht erst beachtet. Sehr nützlich ist die Funktion ‘Find’. Innerhalb eines kleinen Umkreises sucht CAD project nach einem End- oder Schnittpunkt. Dessen Koordinaten lassen sich auf eine von drei Funktionstasten legen und stehen für die Eingabe in den Dialogboxen so auf Tastendruck zur Verfügung.

Ferner können die Funktionstasten mit kurzen Texten belegt werden, die z.B. beim Bemaßen zur Verfügung stehen. So spart man sich das lästige Eingeben etwa von Toleranzangaben.

Besonderes Geschick bewiesen die Programmierer bei den Dialogboxen von CAD project. Während sie geöffnet sind, hat man Zugang zu den Desktop- Accessories (siehe Bild2). Wer also Koordinaten berechnen muß, kann dies während der Eingabe mit Hilfe eines Taschenrechner-Accessories tun, ohne die Box zu verlassen.

Der Bildschirm-Fadenkreuzcursor läßt sich genauso wie die Koordinatenanzeige auf Tastendruck ein- und ausschalten. Wen letztere beim Zeichnen an einer bestimmten Stelle stört, kann sie mit den Cursortasten frei auf dem Schirm verschieben.

Im Menürahmen von CAD project findet der Anwender auch Funktionen, die eigentlich aus dem Bereich der Graphik-Programme stammen. So ist es beispielsweise möglich, mit Mustern gefüllte Objekte wie

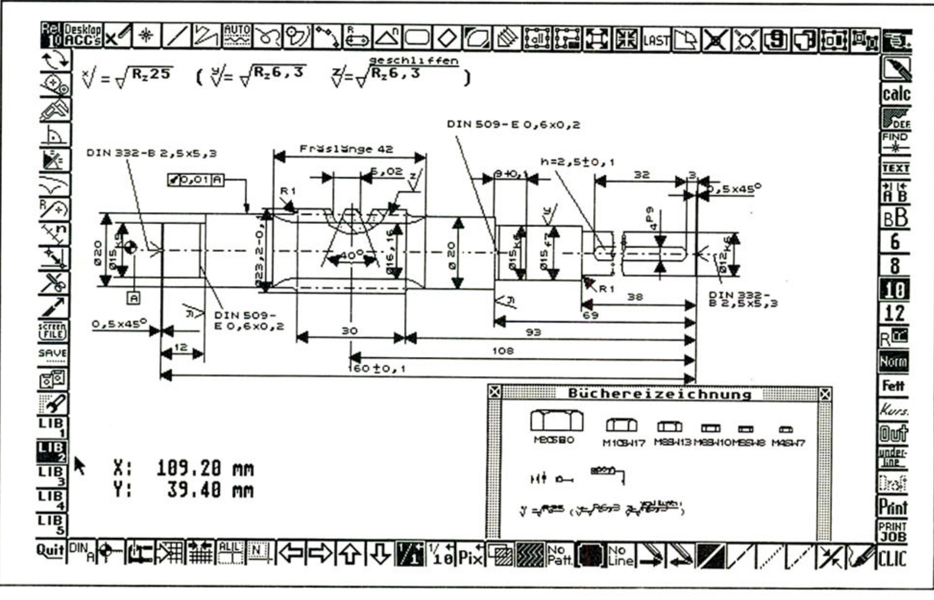


Bild 3: Der Arbeitsbereich mit geöffneten Bibliotheksfenster

geht weiter...

PARAMETERÜBERGABEN

MATRIXDRUCKER
Papier max: DIN A4 | DIN A3
Kopfbauart: 9 | 18 | 24 Nadeln
Init Draft :1B2A270000
Medium:1B2A270000
Final:1B2A280000
Linefeed :1B33180D0A
Medium:1B33180D0A
Final:1C332E0D0A
Minimalfeed:00
Medium:00
Final:1C33010D0A
Dhor:180.00 | Mhor:180.00 | Fhor:360.00
vert:180.00 | vert:180.00 | vert:360.00
End of Printing:00
PLOTTER Stiftzahl: 1 | 2 | 4 | 6 | 8
Init:
O.K. | OK+SAVE | CANCEL

Plotter-Befehlsanpassung
Separator: ; Pen up: PU
Select Pen: SP Pen down: PD
Size: DIN A4 | DIN A3 | DIN A2 | DIN A0
Sel.Paper A3: PS3 xmax: 16158 ymax: 11040
Plot absolute: PA
Steps per mm: 40
Text: LB Text End: 3 (ASCII C.)
Text Hight: SI
Text H.Value Factor: 100 %
Text Angle Italics: SL0.26
Text Angle Normal: SL0
Text Rotate 0:DR1,0 90:DR0,-1
180:DR-1,0 270:DR0,1
End of Plotting: PU;SP0;PA0,0
LINE-STYLES 1:LT 2:LT3,2 3:LT1,0.8 4:LT5,4 5:LT2,2 6:LT6,2

Bild 4: Die Eingabebox für Druckparameter

Kreise oder Rechtecke zu zeichnen. Die Füllmuster können, genauso wie übrigens auch die Linienform, in Dialogboxen ediert werden.

Wie bereits den zahlreichen Vorankündigungen zu entnehmen war, gehen die Textfunktionen über das hinaus, was man gemeinhin von einem CAD-Programm erwartet. Die verwendeten Zeichensätze haben GEM- Format und können (allerdings nur mit einem geeigneten Programm) verändert werden. Die Schrifthöhe sowie der Stil sind edierbar. Ferner ist eine Funktion enthalten, mit der Text stufenlos gedehnt und gestaucht werden kann. Diese Features haben CAD project wohl zu dem Prädikat 'Desktop- Publishing-Qualität' verholfen. Davon ist das Programm jedoch weit entfernt, denn weitere Textfunktionen wie Blocksatz sind nicht enthalten.

sind, können mit verschieden starken Glättungen 'verschönt' werden. Auch an eine ASCII-Schnittstelle wurde gedacht. CAD project ist damit in der Lage, speziell erstellte Dateien zu lesen und aus den darin enthaltenen Koordinaten Polygonzüge herzustellen. So braucht ein eigenes Programm als Ausgabe lediglich eine ASCII-Datei zu erstellen, mit Hilfe derer CAD project dann beispielsweise ein Diagramm zeichnet und auch maßstäblich druckt.

Schließlich sei hier noch eine Funktion genannt, die sicher so manchen Benutzer vor größerem Ärger bewahren wird: das automatische Backup. Nach einer zu bestimmenden Anzahl von Zeichenaktionen wird die momentane Arbeit abgespeichert. Bei größeren Fehlern steht dann immerhin eine relativ aktuelle Version zur Verfügung. Dieses Verfahren empfiehlt sich, denn CAD project hat keine 'Undo'-Funktion. Lediglich das zuletzt gelöschte Objekt kann zurückgeholt werden, nicht aber jegliche Funktion widerrufen werden.

Bibliotheken contra Fleißarbeit Technische Zeichnungen, ob nun am Reißbrett oder mit dem Rechner erstellt, sind mühselig. Diverse Zeichnungs- und Darstellungsnormen behindern die Kreativität des Konstrukteurs. Besonders lästig und zeitraubend wird die Arbeit oft durch die Unmenge von Normteilen wie Schrauben, Kugellager oder im Bereich der Elektrotechnik die Schaltzeichen. Generationen vor uns benutzten Schablonen, um sich dergleichen Mühe zu sparen. Die Schablonen der CAD- Programme heißen Symbolbibliotheken.

Eine solche Bibliothek in CAD project zu erstellen gelingt sofort. Es handelt sich dabei um eine normale Zeichnung beliebiger Größe, die einfach umbenannt wird. Es ist also ohne weiteres möglich, sie während der Arbeit ständig zu erweitern. Ist eine solche Bibliothek erst einmal erstellt, kann sie auf Mausklick geladen und ihr ein Element entnommen werden. Leider verbietet es CAD project, eine solche Zeichnung zu zoomen, also vergrößert darzustellen. Bei kleinen Symbolen muß man daher wissen, wo sie liegen, da sie fast nur noch Punktgröße haben (siehe Bild3).

geht weiter...

Kleine und große Hilfen

Bei technischen Zeichnungen stellt sich recht häufig das Problem, eine gegebene Strecke in gleich große Abschnitte zu teilen. CAD project hat hierfür eine Funktion. Da wohl den Autoren diese zu wenig erschien, bietet sich auch die Möglichkeit, logarithmisch oder im Goldenen Schnitt zu teilen. Das Berechnen von Flächen- und Massenträgheitsmomenten (bei Rotationskörpern) ist genauso problemlos wie das Zeichnen von Interpolationskurven (Splines) nach verschiedenen Algorithmen. Letztere Funktion ist allerdings in der Normalversion eingeschränkt, die erste fehlt völlig.

Freihandlinien, die bei CAD-Programmen, obwohl nötig, leider nicht gang und gäbe

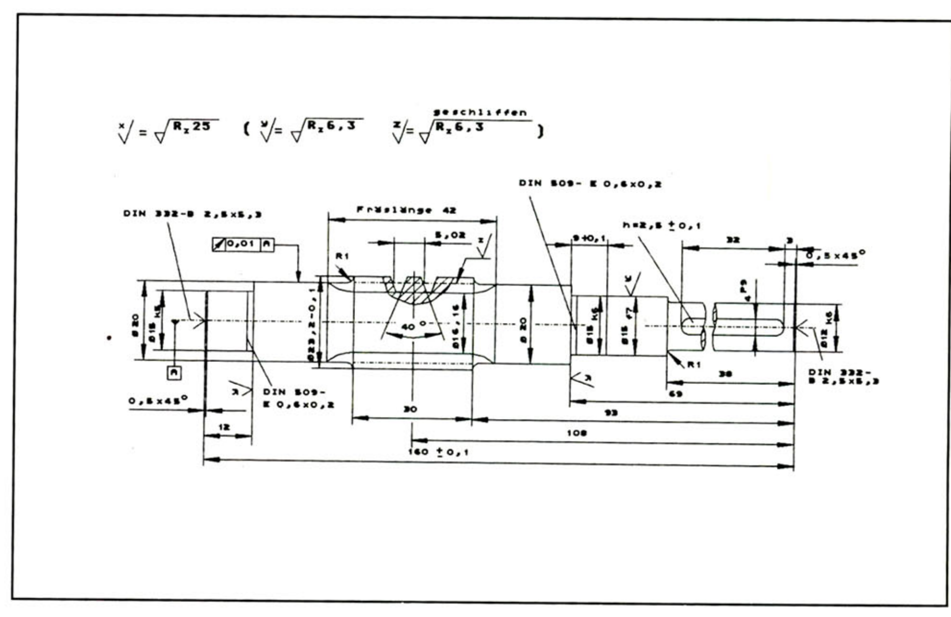
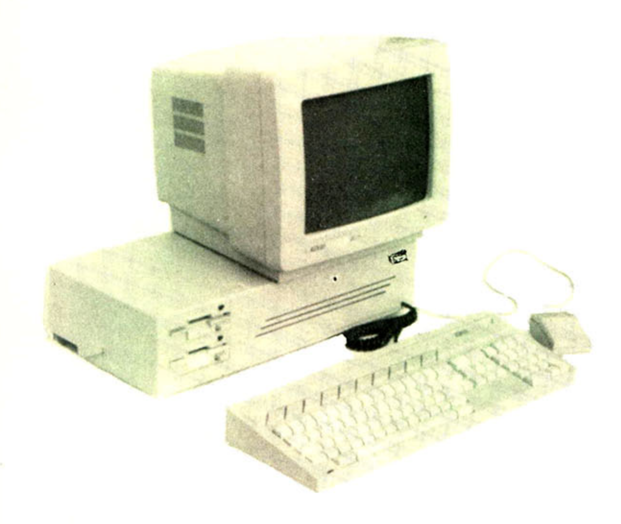


Bild 5: Ein Anwendungsbeispiel aus dem Maschinenbau

520 STM *auf dem* NEUESTEN STAND

DAS PC-GEHÄUSE



SPEZIELL FÜR IHREN ATARI 260/520 ST

KOMPAKT-KIT⁺

BAUSATZ 398,00 DM
ANSCHLUSSFERTIG!
NUR 1.298,00 DM

KOMPAKT-KIT BEINHALTET:

★ Flaches, abgesetztes TASTATUR-GEHÄUSE mit RESTKNOPF und voll entstörter Schnittstellenplatine und SPIRALKABEL.

★ Hauptgehäuse ist vorbereitet für bis zu 2 LAUFWERKE UND EINE HARDDISK (Atari und die meisten Fremdhersteller) mit allen dazu benötigten Kabel, Befestigungen und Blenden.

★ SCHALTNETZTEIL (VDE- und Post-zugelassen: versorgt Rechner, Harddisk und Laufwerke. ZENTRALER NETZSCHALTER an der Vorderseite des Hauptgehäuses.

ALS BAUSATZ
KOMPAKT-KIT .... 398,00 DM
MIT NEC 1036A .... 598,00 DM

★ Hauptgehäuse wird auf ST-Untergehäuse mit Zwischendeck aufgebaut, sodaß ALLE URSPRÜNGLICHEN SCHNITTSTELLEN BLEIBEN. Der komplette Einbau OHNE LÖTEN – AUSFÜHRLICHE GEBRAUCHS-ANLEITUNG.

ANSCHLUSS FERTIG 1298,00 DM

FERTIG UMGEBAUTER 520STM MIT 1 NEC DOPPELSEITIG. LAUFWERK, MAUS & BASIC. HARDDISK: ZWEITES LAUFWERK: SPEICHERERWEITERUNG, ENTSPRECHENDER AUFPREIS

ATARI MONITOR SM 124 .. 448,00
ATARI HARDDISK SH 205 1198,00
AB JETZT: 1040 KOMPAKT-KIT!!!

HARDDISK-ERWEITERUNGS KIT .... 98,00 DM

★ Benötigtes Kabel und Einbaumaterial für Atari Harddisk. (204)

★ Zeitverzögerungsschaltung: Gewährleistet gemeinsames Anschalten von Harddisk und Rechner über zentralen Netzschalter.

★ Akku-Pufferung für die Uhr innerhalb des Tastaturprozessors (Akkus extra).

SCHALTNETZTEILE ab 118,00 DM
AZTEK (VDI- & POST-ZULASSUNG)

LAUFWERKE .... 238,00 DM
NEC 1036A 3,5 DOPPELSEITIG 1 MBYTE

TASTATURGEHÄUSE . 128,00 DM

★ Flaches, abgesetztes TASTATUR-GEHÄUSE mit RESETKNOPF und voll entstörter Schnittstellenplatine und SPIRAL-KABEL.

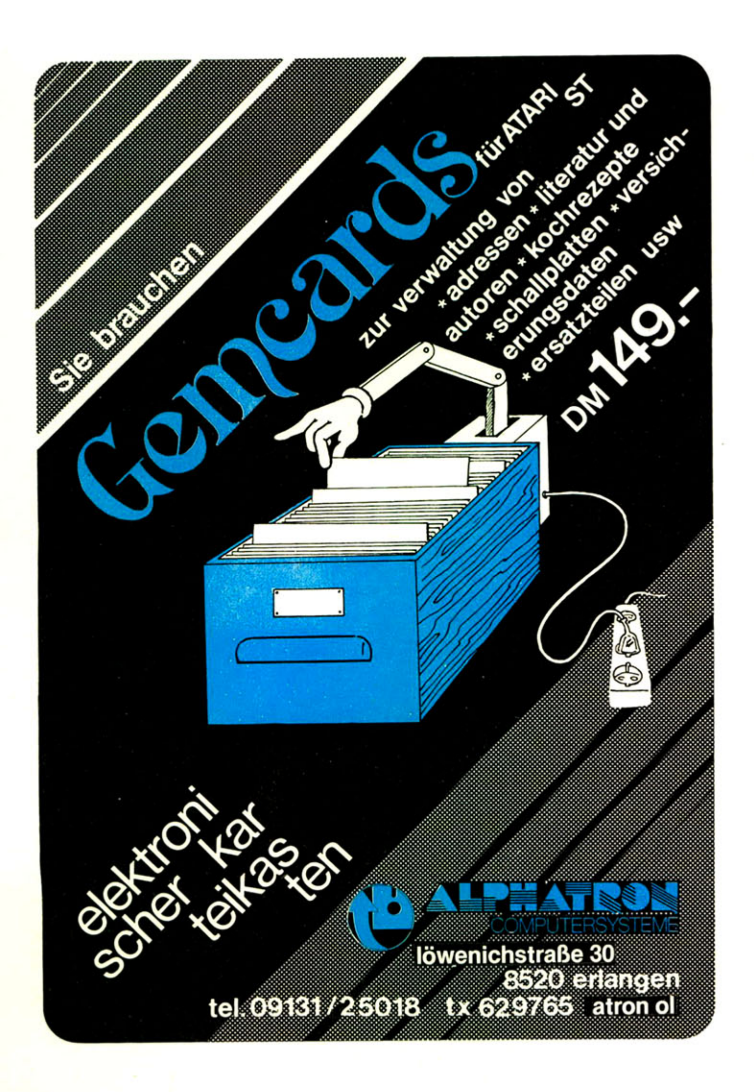
20, 40 & 60 MBYTE HARDDISKS & STREAMERS: MONITOREN & SPEICHERERWEITERUNGEN!!

DISKETTENSTATIONEN
N E C 1036A 3,5" DOPPELSEITIG 1 MB in Gehäuse mit Stromversorgung. Voll ATARI kompatibel, Anschlussfertig.

EINZELSTATION ..... 348,00 DM
DOPPELSTATION ..... 648,00 DM



RIEDSTR. 2
7100 HEILBRONN
Telefon 07131 78480
Telefax 07131 79778



Software zum fairen Preis

Signum!Zwei	DM 369,-
STAD 1.2	DM 158,-
Megamax Modula-2	DM 379,-
GFA-BASIC Interpreter V2.0 oder Compiler	DM 88,-
GFA-ARTIST	DM 128,-
GFA-MOVIE	DM 129,-
GFA-DRAFT plus	DM 298,-
Tempus V 2.00	DM 99,-

Weitere Software: Preisliste anfordern (frankierten Rückumschlag beilegen)!

Bei Vorkasse (Euroscheck beilegen) keine Versandkosten!
Bei Nachnahme (nicht ins Ausland) zuzüglich DM 5,-
Auslandsversand nur gegen Vorkasse zuzüglich DM 10,-

Wir sind vom 19. März bis zum 06. April in den Ferien!

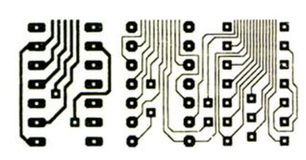
RIA BEERS Software-Vertrieb
Goddardstraße 42, 5300 Bonn 1

ATARI ST
NEC PINWRITER P5/P6/P7

PLATINEN~LAYOUT

MPK Printed Circuit Board Editor. Version 3.0 new!

Professionelles, voll mausgesteuertes Leiterplatten-CAD-Programm ohne Autorouter. Hardwarevoraussetzung: ATARI 1040 STF mit einem NEC 24Nadel PINWRITER (auch MEGA ST möglich).
Platinen bis 203 • 240mm, beidseitig, 1/180 Inch Auflösung. Semifeinstleiter (zwei zw. benachb. IC-pins), 45°Grad.
Gesamtbildschirm-Fadenkreuzcursor mit einstellbarer Schrittweite.
WYSIWYG und ein sehr schneller Bildaufbau erleichtern die interaktive Layouterstellung (Progr. läuft nur monochrom).
Jederzeit TOP VIEW oder BOTTOM VIEW (Spiegeln).
Ausdruck mit hoher Auflösung 1:1 für Prototypen oder 2:1 in Produktionsqualität (!garantiert! Sie ersparen sich den Photoplotservice!).
Ausfüllen der Lötaugenmittelpunkte beim Drucken möglich.
Getrennter Bohrplanausdruck (Lötstopmaske).
MPK PCB EDITOR mit deutsch. Handbuch DM 149.- zuzgl. Vers. per Nachnahme.
Originalausdruck:



MPK
Marek Petrik
Vogelsbergstr. 13
3550 Marburg 7
06421/47588

Händlerkonditionen auf Anfr.!

Drüber und drunter

Vom Ebenenbegriff, wie man ihn von der klassischen CAD-Software her kennt, weicht CAD project ab. Im allgemeinen ist es möglich, mehrere Zeichnungen quasi transparent übereinander zu legen. So können, auf mehreren Ebenen verteilt, die Teile einer Übersichtszeichnung ausgegeben werden. Gearbeitet wird dann auf jeweils einer aktuellen Ebene. Das dient auch der Geschwindigkeit, denn dann braucht das Programm nur die Elemente eben dieser einen Ebene zu berechnen.
Im Gegensatz dazu kennt CAD project nur den Begriff der Hintergrundzeichnung. Eine beliebige Zeichnung wird in den Hintergrund geladen und ist sichtbar, eine Bearbeitung jedoch unmöglich. Ein Trost ist, daß man bis zu vier Zeichnungen nebeneinander und voneinander unabhängig bearbeiten kann, jedoch dürfen diese zusammen nicht größer als DIN A0 sein.

Das Maß der Objekte

Beim Bemaßen einer CAD project-Zeichnung stehen Voll- und Halbautomatik zur Verfügung. Bei der Normalversion muß man sich mit letzterer begnügen, d.h. die Maßhilfslinien wie die Maßlinie selber müssen 'zu Fuß' gezeichnet werden. Nur die Maßzahl wird vom Programm geliefert und mit der Maus positioniert. Die Vollautomatik erstellt die genannten Linien selbst. Natürlich können auch Winkel und Kreise bemaßt werden, allerdings nicht zwei beliebige Punkte der Zeichnung, wie z.B. Eckpunkte einzelner Objekte.
Für Bemaßungen gibt es in der Praxis die verschiedensten Normen. Teils sind sie veraltet, teils sind es Werksnormen. Warum sich CAD project bei diesem Durcheinander nicht an die allgemeingültige DIN-Norm hält, ist unverständlich. Diese schreibt z.B. vor, daß die Maßzahlen auf der Maßlinie stehen müssen und diese nicht unterbrechen dürfen, wie es CAD project automatisch tut. Einen Widerspruch zur Norm stellt auch die Schriftform selber dar. Insbesondere ihre Höhe bei der Druckausgabe folgt nicht den Vorschriften der DIN.

In diesem Zusammenhang fällt jedoch positiv auf, daß die Stufung der Strichstärken normgerecht ist, wenngleich diese natürlich nur auf einem Plotter korrekt wiedergegeben werden können.

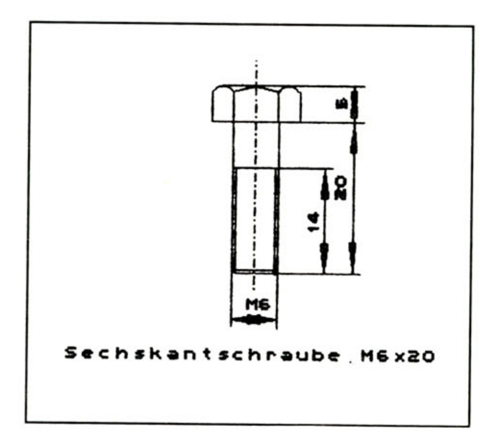


Bild 6: Gedruckt in maximaler Qualität mit 9 Nadeln

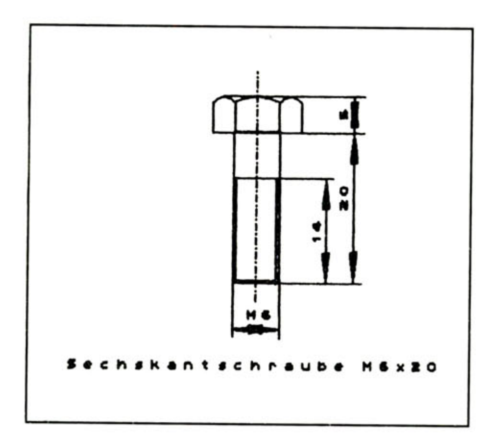


Bild 7: Draft Qualität - gedruckt mit 24 Nadeln

Die Zeichnung kommt aufs Papier

Ein CAD-Programm dient dazu, eine Zeichnung rationell zu erstellen und sauber zu Papier zu bringen. Daher ist einer der wichtigsten Gesichtspunkte die Qualität des Druckes. CAD project ermöglicht die Ausgabe auf graphikfähigen Druckern mit 9, 18 oder 24 Nadeln sowie HPGL und ähnlichen Plottern.
Die Druckausgabe auf Matrixdruckern ist sehr sauber (siehe Bilder), was vor allem für Besitzer der Normalversion wichtig ist, denn sie verfügen über keinen Plottertreiber. Die Druckeranpassung ist komfortabel und einfach, bei der Ausgabe stehen drei Qualitäten zur Verfügung (siehe Bild). Leider unterstützt das Programm nicht die 24-Nadler, die vertikal nur über eine Auflösung von 180 Punkten pro Zoll verfügen. Mit diesen muß auf Draft-Qualität zurückgegriffen werden.
Wenn mehrere Zeichnungen in Folge geplottet werden sollen, können diese in Plotfiles abgelegt werden. Die Ausgabe dieser Dateien kann dann zu einem späteren Zeitpunkt direkt aus der Dateiauswahlbox erfolgen. Es ist jedoch nicht möglich, eine Zeichnung auf einem Drucker in mehreren Teilen auszudrucken, um diese später wieder aneinanderzusetzen. Hier zeigt CAD project seinen professionellen Anspruch, den aber die meisten Heimanwender mangels Finanzkraft nicht erfüllen können.

Arbeiten mit CAD project

Software und Rechner allgemein dienen dem Anwender als Werkzeug. Egal, ob Texte oder Zeichnungen erstellt oder Berechnungen durchgeführt werden müssen; wesentlich für die Leistungsfähigkeit des Werkzeugs sind weniger extravagante Funktionen als die einfache und übersichtliche Bedienbarkeit. Menüs und ihre Form sind ein ewiger Streitpunkt und letztlich Geschmacksache. Den Programmierern von CAD project ist jedoch ein praxisorientiertes Produkt gelungen. Das liegt nicht zuletzt an der engen Zusammenarbeit mit Endanwendern.
Die kleinen Symbole des Menürahmens erschließen sich nach einmaliger Lektüre des Handbuchs und sind dann auch nach längeren Arbeitspausen eindeutig wiederzufinden. Durch die Belegung der Maustasten mit den wichtigsten Funktionen läßt sich mit CAD project rationell arbeiten. Vor allem das Entwerfen eines Bauteils, dessen Form noch nicht völlig bekannt ist, wird durch den schnellen Wechsel in den Ediermodus erheblich erleichtert. Denn wer auf diese Weise am Bildschirm konstruiert, wird sehr häufig mit der Notwendigkeit konfrontiert, Teile der Zeichnung rotieren, vergrößern oder kopieren zu müssen.
Besonders gelungen ist die bereits erwähnte Belegung von Tasten mit Koordinatenpaaren wie auch das Verschieben des Bildschirmausschnitts. Es ist nämlich nicht nur möglich, ganze Seiten weiter zu 'blättern'; die Scrollweite kann auch eingegeben werden.
Leider verfügt CAD project nicht über einen Fangradius. Ein solcher dient dazu, beispielsweise beim Zeichnen einer Linie in einem gewissen Umkreis um ihre Endpunkte nach anderen Eck- oder Endpunkten in der Zeichnung zu suchen und ggf. diese anzuspringen. Wahrscheinlich ist das der Grund, warum sich beim Zoomen eines Ausschnitts relativ starke Ungenauigkeiten, hervorgerufen durch interne Rundung, bemerkbar machen. Der Zoom ist (noch) nicht stufenlos, genausowenig wie die Textdarstellung.

geht weiter...



– Software die es in sich hat!

Nun auch in Deutschland!

Schwarz auf Weiß kann jetzt jeder auf seinem ATARI ST unter MS-DOS arbeiten. Doch nicht nur monochrom, nein, auch in Farbe ist nun der Zugriff auf die Welt der PC-Rechner möglich.

Die Software-Emulation **PC ditto** öffnet allen ATARI ST Anwendern das Tor zum gewohnten professionellen Business Standard.

Mit dem **PC ditto** können Sie mühelos mit Lotus 1 – 2 – 3 oder Symphony Ihre Kalkulationen erstellen.

Mit dem **PC ditto** können Sie Ihre Daten mit DBase III plus verwalten.

Mit dem **PC ditto** haben Sie Zugang zu dem schnellen Turbo Pascal Compiler.

Mit dem **PC ditto** läuft Ihr GW BASIC Interpreter, aber auch Borlands neuestes Kind Turbo Basic.

Mit dem **PC ditto** läuft auch die Software, die es für den ATARI ST noch gar nicht gibt.

Mit dem **PC ditto** laufen so viele Programme, daß wir sie hier gar nicht alle auflisten können.

Der **PC ditto** unterstützt die ATARI-Festplatte, den Druckerport, sogar den Laserdrucker und alle Schnittstellen, soweit es die Hardware des ST zuläßt.

Machen Sie aus Ihrem ATARI ST den preiswertesten PC-Clone! Bestellen Sie den **PC ditto** zum Super-Preis von nur DM 198,–.

Wir laden Sie ein! Steigen Sie mit Ihrem ATARI ST in die Welt des MS-DOS ein.



Hiermit bestelle ich

_ *PC ditto für DM 198,–*

Anruf genügt!
Tel.: 0 61 96/48 18 11, Mo–Fr 9–13 und 14–17 Uhr.
Schriftliche Bestellung nur gegen Vorauskasse oder Nachnahme (Versandkosten DM 7,50; bei Nachnahme zuzüglich DM 3,50 Nachnahmegebühr)

Name: ______________ Vorname: ______________

Straße: ______________ Ort: ______________

Unterschrift: ______________

Nähere Informationen gegen ausreichend frankierten Rückumschlag nur bei:



Industriestraße 26
6236 Eschborn

MS-DOS und GW-BASIC sind Warenzeichen von Microsoft Corp. Lotus 1 – 2 – 3 + Symphonie sind Warenzeichen von Lotus Dev. Corp. Turbo Pascal und Turbo Basic sind Warenzeichen von Borland Corp. DBase III Plus ist ein Warenzeichen von Ashton-Tate Corp.

Seltsamerweise benutzt CAD project die GEM-Fonts, die bekanntlich nicht beliebig zu vergrößern und verkleinern sind. So kann ein Text nur im Normalzoom eingegeben werden, bei einer größeren Zeichnung eine schwierige Angelegenheit, denn Einzelheiten sind dann kaum noch zu erkennen und ein Positionieren des Textes unmöglich. Dafür muß der betreffende Ausschnitt erst wieder vergrößert werden. CAD project bedient sich noch weiterer GEM- Funktionen, die das Bild trüben. Das sind zunächst die Pfeillinien. Auf dem Bildschirm werden die GEM- Pfeile benutzt, die wiederum nicht stark verkleinert werden können. Die Maßlinien erscheinen daher im Normalzoom oft als schwarze Flecke. Einfach ist die Handhabung des Schraffierens von Flächen, die durch Linien mehrerer Objekte begrenzt werden. Leider wird (wiederum nur auf dem Monitor) die systemeigene Füll-Funktion benutzt. Beim Zoomen eines Objektes kann es passieren, daß sich durch Rundungsfehler diese Kontur öffnet. Es läuft dann die Schraffur, wie von Malprogrammen bekannt, aus.

Einer der interessantesten Punkte an CAD project ist die Tatsache, daß es in GFA-Basic geschrieben wurde und so lautet natürlich eine der brennendsten Fragen: Wie steht es um die Geschwindigkeit? Die Antwort ist einfach, denn CAD project braucht sich auch hier nicht hinter der Konkurrenz zu verstecken.

Der subjektive Eindruck ist gut, wenn auch bei Funktionen, die ein häufiges Darstellen des Bildschirms erfordern, leichte Einschränkungen gemacht werden müssen. Das sind vor allem der Fadenkreuzcursor, die Koordinatenanzeige und Funktionen wie das Bemaßen, wo die Maßzahl 'an der Maus hängt'. Dabei tritt teilweise störendes Flackern auf. Die Druckausgabe ist jedoch schneller als bei vergleichbaren Produkten. Mit einem 24-Nadel-Drucker braucht man auf einen Testausdruck in Draft-Qualität kaum zu warten.

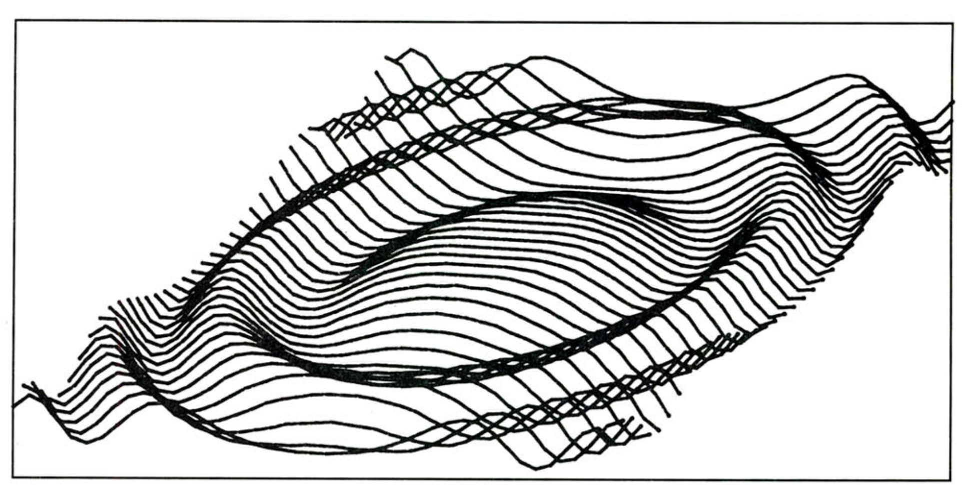


Bild 8: Eine Demonstration des Hidden-Line-Modus im Plotterbetrieb. Die Kurven wurden als Polygonzüge über die ASCII Schnittstelle eingelesen

Fazit

Mit CAD project erwirbt man ein durchdachtes Produkt, das vor allem durch einfache Bedienung und eine Reihe von Sonderfunktionen hervorsticht. Fehler, die noch in der ersten Version enthalten sind, versprechen die Autoren zu beheben. Es ist zu hoffen, daß Schwarzstein weiterhin 'Modellpflege' betreibt und weitere Funktionen implementiert. Aber auch bei den vorhandenen gibt es durchaus noch Handlungsbedarf, so z.B. Bemaßen und bei Textdarstellung.

Wer das Konzept bereits von anderen Programmen kennt, wird kaum Einarbeitung benötigen, dem Neuling wird diese leicht fallen. Wer von einem CAD-Programm auch einige Graphikfunktionen erwartet, der wird sich erst recht mit CAD project anfreunden. Nicht zuletzt der günstige Preis der Normalversion, deren Fähigkeiten für die meisten Anwendungen ausreichen, spricht für dieses Programm.

Ingo Brümmer

CAD project
Entwickler - Schwarzstein Software
Vertrieb - Philgerma, München
Preis - 298 DM (Normalversion)
- 798 DM (Vollversion)

ENDE

DIE FESTPLATTE (TEIL III)

- Wahr ist, daß der Hostadapter zur Verständigung zwischen Computer und Festplatte aufgrund unterschiedlicher Schnittstellen unbedingt notwendig ist.

- Unwahr dagegen ist, daß ein bayrischer Techniker bei der Namensgebung Pate stand, als er nach seinen Erläuterungen zu dieser Platine die Verständnisabfrage "Host mi?..." aussprach.

Irgendwie erinnert mich die folgende Beschreibung des Hostadapters an die gute alte Zeit, als Herr Tramiel noch Eigentümer der Firma Commodore war, und den ersten billigen Taschenrechner auf den Markt brachte (Tatsache: Der Taschenrechner von Commodore mit vier Grundrechenarten war mit etwa DM 200,- der erste billige Taschenrechner für eine breite Käuferschicht). Damals muß sich sein Motto "Power without the price" zusammen mit seinen konsequenten Inkonsequenzen geprägt haben. Die Geräte, die im Laufe der Jahre von ihm auf den Markt gebracht wurden, waren zum Beispiel mit hochwertigen Schnittstellen ausgerüstet (der PET hatte einen IEEE-Bus - es soll noch heute Labors geben, die damit Datenerfassungen fahren!), hatten aber keinen Norm-Steckverbinder; oder die Geräte der nachfolgenden Generation waren nicht kompatibel zu ihren Vorgängern.

Als ATARI dann von Herrn Tramiel gekauft wurde und der erste ST als Gerücht seine Runde machte, wollte die Fachwelt ihren Ohren und dann auch Augen nicht trauen: 68000-CPU, RAM bis 4 MByte, RS-232-Anschluß, Centronics-Schnittstelle, Standard-Floppyanschluß usw. Da sah man auch gerne über kleine Unzulänglichkeiten hinweg - z.B. über den Floppy-Stecker (Was soll denn das sein ?) - ganz konnte er es wohl doch nicht lassen. Ach ja - und dann noch das Computer-System-Interface. Aus dem genormten "S" für Small wurde mal wieder ein "A" für ATARI. Es ist doch zum Mäusemelken...

Anrede

Bevor ich jetzt auf die Eigenschaften (oder sagen wir besser Eigenarten) dieser Schnittstelle eingehe, möchte ich zum besseren Verständnis noch einmal die Begriffe kurz erläutern. Das ACSI (ATARI Computer System Interface) ist die Schnittstelle an Ihrem ST. An diese 19-polige Buchse können bis zu acht verschiedene Peripheriegeräte (Targets) angeschlossen werden. Die Belegung der Buchse ist im nebenstehenden Kasten dargestellt. Diese Schnittstelle ist dem Small-Computer-System-Interface (SCSI) sehr ähnlich; da diese aber eine Menge (im Moment) unbenutzter Leitungen mit sich führt, wurde sie wohl aus "Tramielschen" Gründen zur einfacheren ACSI umfunktioniert.
Sinnigerweise muß jetzt in der Festplatte diese Schnittstelle wieder auf SCSI regeneriert werden (da sind wohl damals bei ATARI die Herren Entwickler über die eigenen Möglichkeiten gestolpert - ein neuer Festplattencontroller am ACSI-Bus ist doch nicht so einfach...). Beinahe wäre aus dem Hostadapter auch ein vollständiger SCSI-Bus geworden, aber dies wurde noch - oder sollte ich sagen, wieder einmal - verhindert. Warum, das werden Sie weiter unten noch erfahren. Der Adapter sitzt in der Festplatte und ist ein Teil des AHDI (ATARI Hard Disk Interface), der zweite Teil ist die zugehörige Software (AHDI.PRG), die Sie beim Booten über den AUTO-Ordner in den Rechner laden.

geht weiter...

Marconi RB2 Trackerball

Die Maus ist tot, es lebe der Trackerball

Marconi ist ein renomierter Hersteller von Trackerballs mit weltweitem Vertrieb. Die Anwendung des Trackerballs in der Flugsicherung und im militärischen Bereich sprechen für die Zuverlässigkeit des Produkts. Der Trackerball eignet sich hervorragend im **CAD/CAM** Bereich, in der **Textverarbeitung** und zur **Positionierung** des Cursors auf dem Bildschirm.



Der **Marconi** Trackerball ist weit mehr als nur ein Mausersatz! Dank seines kompakten Gehäuses benötigen Sie in Zukunft keinen Platz mehr zum Bewegen einer Maus. Die Drehung des Trackerballs wird digital umgesetzt und entspricht der Bewegung des Cursors auf dem Bildschirm. Die Firma **Weeske** steht mit ihrem Namen hinter diesem Spitzenprodukt und ist Ihr Ansprechpartner für den Vertrieb in Deutschland, der Schweiz und Österreich.

Achtung Computer-Fachhändler

Immer wieder bedauern Kunden am Telefon die schlechte Zubehörversorgung vor Ort. Dieser Umstand treibt sie in die aufgehaltenen Arme der Versandhändler. Sie haben die Möglichkeit diesen Zustand beim Marconi Trackerball zu ändern.

Fordern Sie doch einfach die Einkaufskonditionen mit Mengenstaffel an!

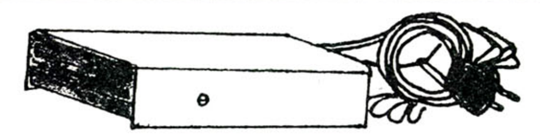
Technische Daten: • Stromversorgung: 5 V +/- 5 % • Spannungsaufnahme: 60 - 200 mA (je nach Ausf.) • Output Signal: TTL • Kugeldurchmesser: 2,25 inches • Gewicht: 0,6 kg • Abmess.: (B/L/H) 115 x 200 x 70 mm.

nur 198,--

Diskettenlaufwerke

(NEC / TEAC) für Atari ST

• Erstklassige Verarbeitung • komplett mit Metallgehäuse, Netzteil und Kabel • leise • anschlußfertig und vollkompatibel •



3,5" Floppy	2x80 Spuren	(720 KB)	**348,--**
3,5" Doppelfl.	4x80 Spuren	(1,4 MB)	**648,--**
5,25 Floppy	2x80 Spuren	(720 KB)	**448,--**
3,5"+5,25" Mix	4x80 Spuren	(1,4 MB)	**798,--**

Einzelteile als Bausatz:

Netzteil	39,--	Gehäuse	29,80
Kabelsatz einfach	29,--	Kabelsatz doppelt	39,80
3,5" Laufwerk einf.	248,--	5,25" Laufwerk einf.	280,--

APPLICATION SYSTEMS  Software

Signum 2
Super Textverarbeitung für unterschiedlichste Anwendung. Für Literaten, Musiker, Sprachwissenschaftler, Physiker, Chemiker, Mathematiker usw.
Viele zusätzliche Fonts sind erhältlich!
448,–

Signum-Fontdisketten

Fontdiskette Julia	100,--
Eurofont Diskette	69,--
Professional Fontdisk	100,--
Fontdiskette Rokwel	100,--
Signum!Utility	89,–

STAD
Das Grafik-Programm, das fast keine Wünsche offen läßt. Neue Version: 99 Bildschirme beim Mega-ST. Laserdruckertreiber für Atari-Laser.
178,–

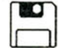
Megamax Modula-2
Modula-2 Entwicklungssystem mit GEM-Shell.
Demodisk (20,--) **398,--**

Megamax C-Compiler
Das bekannte C-Entwicklungssystem! Mit Single Pass-Compiler, Inline Assembler, Disassembler, Linker & Librarien, Vollständige GEM-Libraries, GEM-Editor, GEM-Shell, 370 seitiges Handbuch.
Neuer Preis: **398,–**

Deutsches Handbuch **49,--**

Imagic Der Grafikcompiler
Die Sprache, die Bilder schreibt!
498,--

Creator Demo (20,--)	
FlexDisk Ramdisk	**69,–**
Harddisk Utility	**69,--**
Editor Toolbox	**149,--**

Kieckbusch  Software

Neu!!
Timeworks Desktop Publisher
• mit allen wichtigen Hilfsmitteln, die ein professionelles DTP-System ausmachen • mit Druckertreiber für Laserdrucker • deutsche Version mit ausführlichem deutschem Handbuch •
nur 389,--

VIP Professional 1.4
• Kalkulation • Grafik • Datenbank
Daten- und Bedienungskompatibel mit LOTUS 1-2-3. **Jetzt mit NEC-P6-Treiber!** Kompl. in deutsch, inkl. Hotline u. Updateservice. Mit 3500 verkauften Paketen in Deutschland das meistbenutzte Kalkulationsprogramm
498,--

Programmknüller: **STEVE** **398,–**
Integriertes Programm: Text • Grafik • Datenbank
Programm in Maschinensprache, daher enorm schnell.
Text: Lexikon für Deutsch, Englisch, Russisch und Jugoslawisch verfügbar. Übersetzt wörtlich Deutsch-Englisch. 8 Schriften. Serienbriefe mit Adressen aus der Datenbank.
Grafik: Zeichenmöglichkeiten ähnlich Easy Draw. Zeichnungsübergabe an die Textverarbeitung.
Datenbank: Sehr schnelles Sortieren. Über 20 Datenbank-Funkt.

STEVE-Lehrbuch
Wichtige Informationen über STEVE inkl. Beispiel-Disk. 250 Seiten, 480 Gramm.
58,--

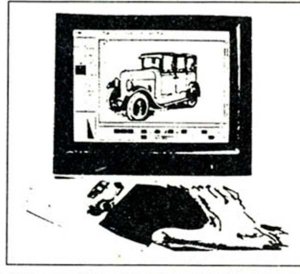
Desk Assist II+
Rundumprogramm: Terminplanung, Kalender, Uhr, Alarmtermine, Notizblatt, Rechner, residente Adressen- und Telefondatei, mit komfortablen Such- und Druckmöglichkeiten. Super-Terminalprg. im Lieferumfang, deutsch **139,--**

A-MAGIC Turbo Dizer
Non plus Ultra unter den **Videodigitizern.** Echtzeitverarbeitung in 16 Graustufen.
298,--

Multi-Hard-copy
Anpassung an wirklich jeden Drucker, ob Matrix oder Laser-Drucker.
98,--

Handy-Scanner für Atari ST

Endlich ist die Grafikeingabe so leicht, daß sie ein wirkliches Vergnügen ist!



- Einsatz im Desktop-Publishing
- Abrastern von Bildern (f oder sw)
- Speichern von Unterschriften
- Anfertigen von illustrierten Handbüchern
- Überarbeiten und Entwickeln von Logos und Typen
- Abbildung auf dem Bildschirm erfolgt in weniger als 3 Sekunden

Das Grafikpaket **CAMERON HANDY PAINTER 2.0** ist im Lieferumfang enthalten! (Bei Atari nur s/w-Bildschirm).
Für Techniker, Lehrer, Werbefachleute, Redakteure ...
Mit deutschem Handbuch! **Info anfordern** (Computertyp)

• Schwarz-Weiß Version **798,--**

• Neu! **Typ 3** mit Graustufen **nur 848,--**

GFA-Systemtechnik **Software + Literatur**

Bücher: GFA-Basic (Ostrowski) **79,–** GFA Handbuch TOS&GEM **49,--**
GFA Basic Programmierung inkl. Diskette **49,–**

GFA VEKTOR
3D-Grafik-Toolbox zum GFA-BASIC
99,--

GFA BASIC Comp.
Schneller 2-Pass-Compiler, benötigt keinen Linker, voll kompatibel zum Interpreter, einfachste Bedienung.
99,–

GFA MOVIE
Zeichenprogramm, Zeichensatzeditor, Drehbuchprogr., Effektprogr., SnapShot Möglichkeit
149,–

GFA STARTER
Startet GEM-Prg. aus dem Auto-Ordner, bis zu 10 Prg. hintereinander startbar.
59,--

GFA BASIC Interpr.
Kompakter Hochgeschw.-Interpreter, Strukturiertes Programmieren, Komfortabler Editor.
99,–

GFA ARTIST
Erstellen eines Films durch Zusammensetzen verschied. Animationen, Arbeitet mit 1024 Farben gleichzeitig.
149,–

monoSTar PLUS
hervorragendes monochromes Zeichenprogramm mit vielen Kniffs und Tricks.
149,--

GFA Objekt
3D-Zeichenprogramm, vollständig in GFA Basic geschrieben, Schnittstelle zu GFA-BASIC,-DRAFT,-VEKT.
198,-

GFA Publisher
Desktop Publishing Prg., Leistungsfähiger Texteditor, Automatischer Textfluß und -umbruch, Silbentrennung.
ab April 88 **398,–**

ST DIGI-DRUM
Drum-Maschine für Atari ST Von 2-Bit-Systems.
79,--

GFA DRAFT
Leistungsfähiges CAD-Prg.
198,--

GFA DRAFT plus
noch komfortabler **349,--**

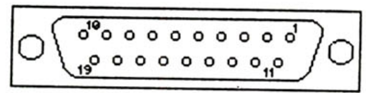
GFA BASIC 68881
Läuft nur mit Motorola 68881 Floating-Point-Coprozessor, Kompakter Hochgeschw.-Interpr. und 2-Pass-Comp.
349,–

Der ACSI-Bus

Das ATARI-Computer-System-Interface ist mit einer 19-poligen D-Subminiatur-Buchse ausgestattet. Von ATARI werden daran z.Zt. die Festplattenlaufwerke und der Laserdrucker angeschlossen. In naher Zukunft wird wohl noch der CD-ROM-Spieler dazukommen. Das obige Anschlußbild zeigt die Draufsicht auf die Buchse am Computer bzw. an den Peripheriegeräten. Die Pins sind im einzelnen wie folgt belegt:

Pin	Belegung	Richtung ST - Periph.
Pin 1 -	Datenbus Bit 0	<--->
Pin 2 -	Datenbus Bit 1	<--->
Pin 3 -	Datenbus Bit 2	<--->
Pin 4 -	Datenbus Bit 3	<--->
Pin 5 -	Datenbus Bit 4	<--->
Pin 6 -	Datenbus Bit 5	<--->
Pin 7 -	Datenbus Bit 6	<--->
Pin 8 -	Datenbus Bit 7	<--->
Pin 9 -	$\overline{CS}$=Chip Select (Klingelleitung...)	---->
Pin 10 -	$\overline{INT}$=Interrupt (Die Quittung für Daten)	<----
Pin 11 -	Masse	
Pin 12 -	$\overline{RESET}$ (die Notaus-Taste)	---->
Pin 13 -	Masse	
Pin 14 -	$\overline{ACK}$=Acknowledge (Annahme der Bestellung $\overline{DRQ}$)	---->
Pin 15 -	Masse	
Pin 16 -	A1 (Fertigmachen zur Befehlsausgabe)	---->
Pin 17 -	Masse	
Pin 18 -	R/$\overline{W}$=Read/Write (Lesen oder Schreiben)	---->
Pin 19 -	$\overline{DRQ}$=Data Request (Daten haben will...)	<----

(Bei einem Querstrich über einer Bezeichnung ist das Signal Low- Aktiv - es geht auf Nullpegel, wenn es aktiviert wird.)



Satire

Apropos acht Peripheriegeräte - die bisher bekannten sind einmal die Festplatte und dann hat sich noch der Laserdrucker an diese Schnittstelle gemogelt. Die Adresse der jeweiligen Geräte wird über kleine DIP-Schalter (Dual in Line Package) eingestellt. Dieser DIP-Schalter befindet sich in der Festplatte auf der Platine des Hostadapters (bei der SH 205 auf der Platine des Controllers, da hier Adapter und Controller zusammengefaßt sind). Der SCSI-Bus hat ebenfalls die Eigenschaft, bis zu acht Geräte ansteuern zu können. Die Auswahl erfolgt über drei Jumper auf der Controllerplatine - 8 Geräte am ACSI-Bus x 8 Geräte am SCSI-Bus = 64...(merken). Wie Sie in der letzten Folge erfahren haben, kann der Adaptec-Controller bis zu zwei Laufwerke ansteuern - 2x64=128... - ach so, Sie haben schon einen Laserdrucker, das ist natürlich Pech, dadurch entfällt ja ein Gerät an dem ACSI-Port - dann müssen Sie sich eben mit 7x8x2=112 Festplatten zufrieden geben.

'Hart'ware

Bei der nachfolgenden Beschreibung des Hostadapters beziehe ich mich auf die SH-204, da er hier noch separat aufgebaut ist. In der SH-205 fehlen die Puffer-ICs zwischen Adapater und Controller, außerdem wurde er um einen gepufferten DMA(ACSI)-Port erweitert.

Im Bild sehen Sie den Bestückungsplan des Hostadapters. Auf der kleinen Platine ist auf der einen Seite die 19-polige Sub-D-Buchse (Kürzel für D-Subminiatur-Steckverbinder, das D bezieht sich auf die Gehäuseform) für den DMA-Port angeordnet. Auf der anderen Seite befindet sich der 50-polige Pfostenstecker für den SCSI-Anschluß. Rund um den PAL-Chip (kein Hundefutter) sind einige Puffer- und Gatter-ICs angeordnet. Was sich hinter dem Begriff PAL verbirgt, finden Sie nebenstehend erklärt.

Auf der Adapterplatine der ATARI-Festplatte kommt ein PAL16L8 zum Einsatz. Die Kennzeichnung bedeutet, daß dieses PAL über die Programmierung maximal 16 Eingänge und 8 Ausgänge haben kann. Wie Sie anhand des Blockschaltbildes erkennen, übernimmt das PAL eine dominierende Rolle in der Ablaufsteuerung des Hostadapters - und hier liegt auch die Besonderheit dieses Interfaces. In der Beschreibung des Funktionsablaufes ist die Kommandostruktur zur Daten- und Steuerungsabwicklung genau festgelegt:

1. Command-Phase
Hier wird über einen sechs Byte-großen Befehl als erstes das auszuführende Kommando vom Initiator zum Target (Ziel) übertragen.

2. Data-In/Out-Phase
Hier erfolgt die eigentliche Datenübertragung.

3. Status-Phase
Hier erfolgt die Rückmeldung, in der die ordnungsgemäße Erledigung des Kommandos oder ein Fehlerkode an den Initiator gemeldet wird.

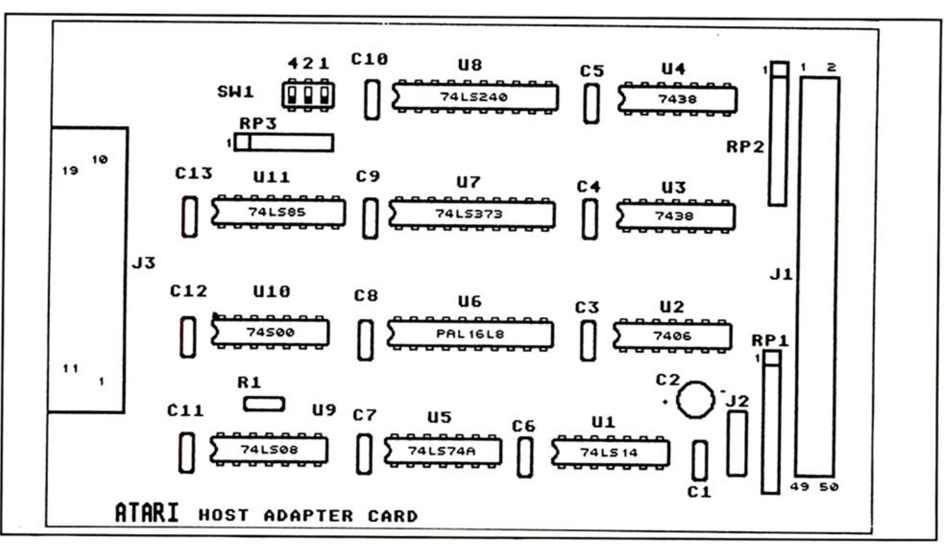


Bild 1

geht weiter...

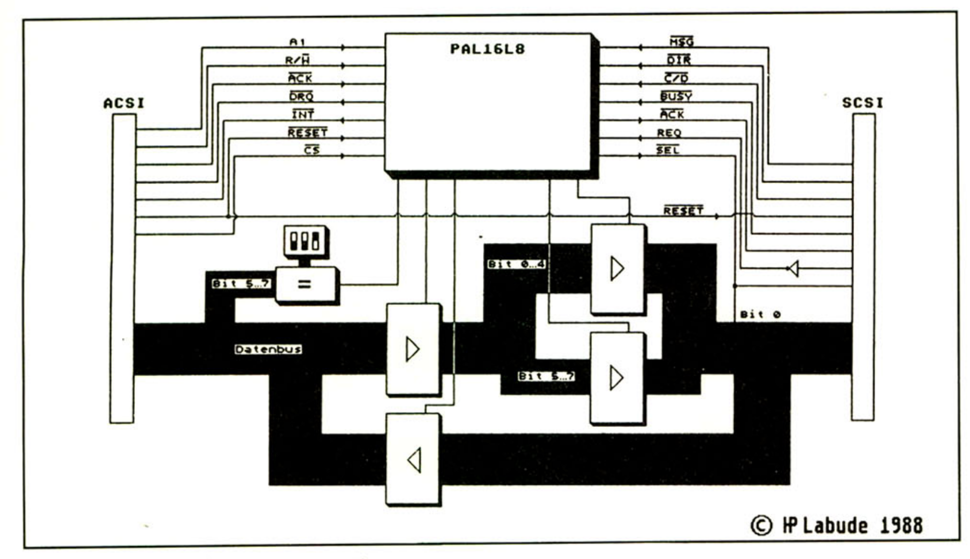


Bild 2: Blockschaltbild des ATARI-Adapters

Danach wiederholt sich der Ablauf von Punkt eins bis drei für jeden zu übermittelnden Datenblock. Woran erkennt nun ein Target, daß es Empfänger für das am Datenbus anliegende Signal ist und um welche Art von Signal es sich handelt?

Fahrplan

Beim Betrachten der ACSI-Anschlußbuchse und der Signale des SCSI-Bus sind Ihnen bestimmt die verschiedenen Steuerleitungen aufgefallen. Durch einen zeitlich genau festgelegten Ablauf der Signale über diese Anschlüsse kann das Target ein Kommando vom Daten- oder Statussignal unterscheiden. Nur sind die Signalleitungen von ACSI und SCSI nicht identisch. Die notwendige Konvertierung übernimmt der Hostadapter. Er setzt also die Signale vom ACSI über Logikschaltungen (in der Hauptsache im PAL) in die notwendigen Steuersignale für das SCSI um. Das Prinzip der Steuerung über Command-/Data- und Status-Phase wird dabei nicht verändert!

Pünktlichkeit

Die Zeitdiagramme der einzelnen Phasen sind im nebenstehenden Bild zu sehen. Wenn man sich beim Betrachten der Diagramme noch die Geschwindigkeiten für die Datenübermittlung über den ACSI-Bus in Erinnerung ruft, wird klar, warum ATARI das Anschlußkabel zur Festplatte so kurz gehalten hat. Jede kleinste Verzögerung durch ein überlagertes Störsignal

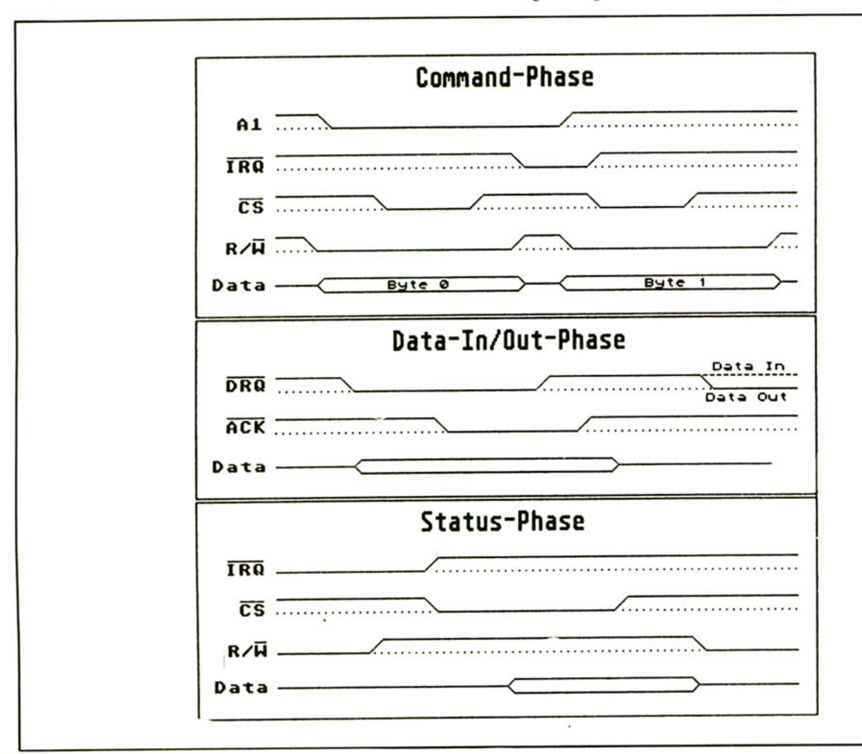


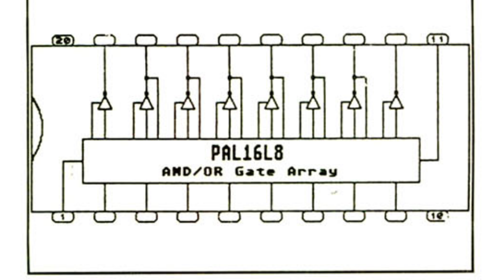
Bild 3: Die Zeitdiagramme der einzelnen Phasen

Was ist ein PAL?

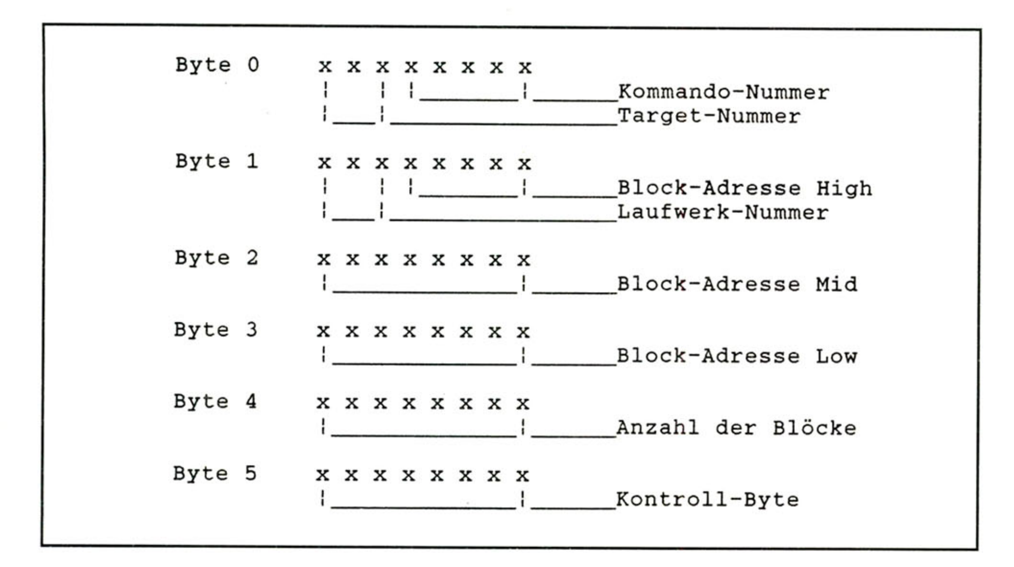
Die Abkürzung PAL steht für Programmable Array Logic, zu deutsch etwa: programmierbares Logik-Feld. Bei den heute gebräuchlichen und auch notwendigen komplexen Schaltungsentwürfen bedarf es einer Vielzahl von logischen Verknüpfungen. Diese werden normalerweise mit herkömmlichen UND-, ODER- und Inverter-TTL-Bausteinen gelöst. Dabei kann aber leicht ein s.g. IC-Grab entstehen, also eine unverhältnismäßig große Anzahl von diesen Bausteinen notwendig sein. In diesem Zusammenhang sollte man sich an den Hardware-Entwurf für die ST-Serie erinnern, das waren vier etwa 50 x 25 cm große Platinen voll mit ICs!

Man kann natürlich jetzt nach Abschluß der Versuchsphase diese ganzen Funktionen in speziell gefertigte ICs packen. Das Beispiel hierfür sind die vier Spezial-ICs in Ihrem ST: GLUE, DMA-Chip, Shifter und MMU. Das lohnt sich nur bei hohen Stückzahlen und hoher Packungsdichte. Für kleinere Lösungen verwendete man am Anfang PROMs, also programmierbare Festwertspeicher. Diese PROMs sind aber für komplexe Funktionen nicht geeignet, da immer nur ein Ausgangszustand durch Anlegen einer Adresse erzielt wird.

Daraus erfolgte die Entwicklung der PALs. Es handelt in erster Linie auch um programmierbare Matrixbausteine, wobei aber im Unterschied zum PROM bereits die logische Funktion programmiert wird. Außerdem kann in der Regel ein Teil der Anschlüsse eines PALs als Ein- oder Ausgang programmiert werden, und es ist zusätzlich eine Rückführung eines Ausgangs in das PAL möglich (Wird ein solcher Ausgang durch die gerade anliegenden Signale ausgeschaltet und damit hochohmig, kann dieser Port als Eingang benutzt werden!). Damit läßt sich aber auch ein solcher Baustein kaum noch kopieren, da man die Funktion als Ein- oder Ausgang nicht mehr erkennen kann. Zusätzlich gibt es eine sogenannte Last-Fuse, nach deren Brennen ein Auslesen verhindert wird (es werden willkürliche Ausgangsmuster erzeugt). Einen besseren Kopierschutz kann man sich wohl kaum vorstellen. Das Anschlußbild des im Hostadapter verwendeten PALs habe ich Ihnen untenstehend aufgezeichnet.



geht weiter...



```
Byte 0     x x x x x x x x
           |   | |_______|______Kommando-Nummer
           |___|_________________Target-Nummer

Byte 1     x x x x x x x x
           |   | |_______|______Block-Adresse High
           |___|_________________Laufwerk-Nummer

Byte 2     x x x x x x x x
           |_____________|______Block-Adresse Mid

Byte 3     x x x x x x x x
           |_____________|______Block-Adresse Low

Byte 4     x x x x x x x x
           |_____________|______Anzahl der Blöcke

Byte 5     x x x x x x x x
           |_____________|______Kontroll-Byte
```

würde ja den ganzen Busfahrplan durcheinander bringen (ein paar Dezimeter mehr sind trotzdem noch drin; davon in der nächsten Folge mehr).

Hausnummer

Jetzt kann zwar die Art der anliegenden Sendung erkannt werden, aber noch nicht der jeweilige Empfänger. Diese Information wiederum steht in der "Mitteilung", die wir uns jetzt einmal näher ansehen.
Das Kommando ist ein 6-Byte-großer Befehl. Er wird Byte für Byte in der Command-Phase übertragen: (siehe Kasten oben
Zum besseren Verständnis des Ablaufes sollten Sie sich jetzt auch die Zeitdiagramme noch einmal näher ansehen. Dem Initiator (also unserem ST) obliegt die alleinige Busverwaltung, da kein Belegt-Signal vorhanden ist. Deswegen können auch verschiedene Geräte am ACSI-Port nicht miteinander kommunizieren (beim SCSI-Bus möglich).

Wenn der Postmann...

Nach Abschluß einer Status-Phase gibt der ATARI den Bus für sich selbst wieder frei. Um nun mit einem beliebigen Gerät in Kontakt zu treten, legt der ATARI das Byte 0 des Kommandoblocks auf den Bus und setzt etwa gleichzeitig die Leitung A1 auf Null (natürlich mit Setzen der R/W-Leitung auf "Schreiben", das können Sie ja aus dem Zeitdiagramm ersehen). In Byte 0 steht in Bit 5 bis Bit 7 die jeweilige Target-"Hausnummer". Jetzt klingelt der ST mit Chip-Select (CS) bei allen Geräten an. Im Blockschaltbild des Hostadapters sehen Sie den DIP-Schalter, mit dem diese "Hausnummer" eingestellt werden kann. Das darunterliegende IC vergleicht die eingestellte Nummer mit der auf dem Datenbus (Bit 5 bis 7) anliegenden. Sind diese identisch, so meldet das PAL es über die Interrupt-Leitung (INT) zurück und der Hostadapter bleibt auf Empfang. Im anderen Fall ignoriert er ganz einfach die folgenden Daten. Um den Bus durch eine Schlafmütze nicht zu blockieren, läßt der ATARI den Targets max. 100 ms Zeit (so schnell sind Sie nicht am "Türdrücker"...), ansonsten veranlaßt er einen Time-Out im System. Wenn die Verbindung steht, werden Stück für Stück die restlichen fünf Bytes übertragen. Dazu sehen wir uns diese im einzelnen an:

Inhalt der Sendung

In Byte 0 steht, wie gesagt, die Adresse des Targets und in den Bits 0 bis 4 die Nummer des jeweiligen Kommandos (sie sind weiter unten einzeln aufgeführt). In Byte 1 steht wiederum im Bit 5 bis 7 die Nummer des Laufwerkes (theoretisch bis zu acht Stück). Die restlichen 5 Bits können zusammen mit den Bytes 2 und 3 eine 21-Bit-Blockadresse bilden. Mit dieser Blockadresse ist jeder Sektor auf der Festplatte ansprechbar, sie errechnet sich nach folgender Formel:

$$Ba = (((Zn * Ak) + Kn) * As) + Sn$$

Dabei bedeuten:
- Ba = Block-Adresse
- Zn = Zylinder-Nummer
- Ak = Anzahl der Köpfe
- Kn = Kopf-Nummer
- As = Anzahl der Sektoren
- Sn = Sektor-Nummer

In Byte 4 finden wir die Angabe über die Anzahl der in der Datenphase zu übertragenden Blöcke und Byte 5 enthält schließlich ein mögliches Kontrollbyte.
Der Inhalt des jeweiligen 6-Byte-Kommandos hängt selbstverständlich vom jeweiligen Befehl ab - was sollte auch die Angabe einer Blockadresse z.B. beim Befehl "Test Unit Ready"? Hier ist nur die Statusmeldung interessant.

Faulpelz...

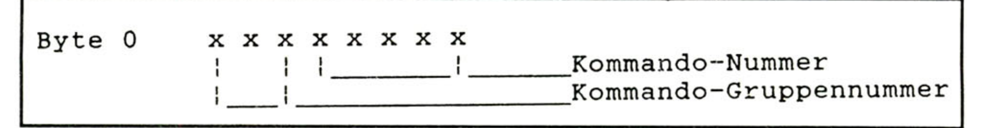
Aus dem Zeitdiagramm ist ersichtlich, daß die Leitung A1 nur während des Bytes 0 zurückgesetzt wird. Für die restliche Übertragung bleibt sie auf Eins (High-Pegel), um eine sonst mögliche Adressierung eines anderen Gerätes zu vermeiden. Die sechs Bytes des Kommandoblocks werden noch einzeln übertragen, da sich der Inhalt der kompletten Nachricht erst aus diesen Daten ergibt.

... und Akkord

In der Daten-In-/Out-Phase geht dann aber richtig die Post ab, denn hier kann der DMA-Chip durch die Übertragung von bis zu 255 Blöcken (á 512 Bytes) in einem Rutsch seine Daseinsberechtigung belegen. Das Ganze wird durch die Statusmeldung abgeschlossen, die unter anderem eine Fehlermeldung enthält. Diese Statusmeldung muß aber in jedem Fall innerhalb von vier Sekunden nach der Command-Phase erfolgen, sonst bricht der ST wiederum mit einem Time-Out die Übertragung ab. Wenn die 255 Blöcke nicht zur Datenübertragung ausgereicht haben, wird der Vorgang - also Command-Phase, Daten-In/Out-Phase usw. - mit den nächsten Daten einfach wiederholt.

Irrtum

Die wenigen Kommandos, die unsere Festplatte vom ST bekommt (der Controller kennt wesentlich mehr), finden Sie im nebenstehenden Kasten erklärt. Dazu möchte ich gleich folgendes ergänzen. In den Orginalunterlagen zu dem AHDI von ATARI werden teilweise falsche Befehle aufgeführt, die der Adaptec-Controller überhaupt nicht versteht. ATARI hat in der Entwicklungszeit der Festplatte mit verschiedenen Controllern experimentiert (unter anderem auch mit einem RLL-Con-



```
Byte 0     x x x x x x x x
           |   | |_______|______Kommando-Nummer
           |___|_________________Kommando-Gruppennummer
```



STARKE SOFTWARE

ST ARCHIVAR
DIE ÜBERZEUGENDE DATEIVERWALTUNG
- Unterstützt die Produktion wissenschaftlicher Texte
- Verwaltet und bearbeitet Zitate u. Literaturangaben und Ausgabe als Text
- Komfortable Nutzung für Video- u. Adressendateien
- Datenverwaltung mit Dateien von bis zu 19 Datenfeldern (38 in Vorbereitung und 200 Datensätzen
- Dynamisch erweiterbar
- Veränderung der Dateimasken
- Speicherung der Daten als DIF oder SDF Datei, damit Übernahme in Datenbank möglich
- Beliebige Programme nachladbar u. auszuführen ohne ARCHIVAR zu verlassen (z. B. 1st Word)
- mit allen Textprogrammen kompatibel, die ASCII einlesen
- Ausdruck von Karten (z. B. Bibliothekskarten)
- Listen-Ausdruck auf Knopfdruck
- Etiketten-Ausdruck
- Einfache Druckerprogrammierung
- Einfach bedienbar mit der Maus

ST-ARCHIVAR enthält zahlreiche Hilfsprogramme
- Bibliotheks-Suchfunktion (ganze Disketten durchsuchen lassen)
- Uhr-Einstellung
- Info-Datei über alle ARCHIVAR-Dateien erstellen
- Druckersteuer-Programm
- Kopierprogramm
- Sortierprogramm
- Erzeugt RAM-DISK G bis 1750 KB
- Automatisches Kopieren in die RAM DISK G beim Start

und das alles für **DM 89,-**

ST PRINT
4 NÜTZLICHE PROGRAMME IN EINEM
RESETFESTE RAMDISK
- Größe einstellbar von 32 – 4000 KB
- kann auf Laufwerk C bis P gelegt werden
- arbeitet problemlos mit einer Harddisk
- sehr schnell, da in Maschinensprache programmiert

DRUCKERSPOOLER,
- Größe einstellbar von 2 – 510 KB
- Arbeitet mit TOS- u. GEM-Programmen
- Hohe Geschwindigkeit beim Ausdruck

DRUCKERVOREINSTELLUNG
- mit der Maus, Knopfdruck statt Handbuch
- viele Einstellmöglichkeiten
- Einstellung des Druckers vom Desktop u. aus jedem GEM-Programm (VIP Prof., Wordplus, Tempus etc.) möglich.

HARDCOPYROUTINE
- nutzt die Fähigkeiten von 9-, 18- u. 24-Nadeldruckern
- versch. Auflösungen, Schnelldruck bis Qualitätsdruck
- Umsetzung der Farben in Grauwerte
- gespoolte Hardcopy etc.

DM 59,-

ST PLOT
KURVENDISKUSSIONS- U. FUNKTIONSPLOTTER-PRG.
- ist ein Programm, das mehr kann, als einfach nur Funktionen zeichnen. Es bestimmt die richtigen Ableitungen u. damit werden Null-, Extrem- und Wendestellen berechnet, es stellt den Definitionsbereich u. die Periode fest und vieles andere mehr.
- zeichnet die Schaubilder von bis zu 3 Funktionen u. deren erste u. zweite Ableitung gleichzeitig auf einen Bildschirm. Danach können Ausschnitte der Funktionen vergrößert und verkleinert werden.
- ist komplett GEM- und mausgesteuert. Die Bedienung ist somit sehr einfach. Sie brauchen nur die Funktion einzugeben, alle anderen Berechnungen erledigt für Sie das Programm.
- läuft sowohl in mittlerer wie auch in hoher Auflösung.
- ist nicht nur für Professoren, Studenten u. Schüler interessant. Es kann überall dort angewendet werden, wo man sich mit Funktionen beschäftigt.
- Umfangreiche Eingabemöglichkeiten, z. B. Verwendung sämtl. Funktionen eines wiss. Rechners. Defin. von 10 versch. Zahlenwerten als Konstanten, e u. Pi verwendbar.
- Komfortable Zeichnung, z. B. autom. Zeichnung u. Beschriftung der Achsen und deren Skalierung, Fehlerroutinen, damit keine unnötigen Asymptoten gezeichnet werden, beliebig viele Funktionen nacheinander in ein Bild einzeichenbar ermöglicht Funktionenvergleich u. Ablesung der Schnittpunkte etc.
- Beste Berechnung der Funktionswerte, z. B. Hohe Rechengeschwindigkeit, hohe Rechengenauigkeit.

DM 59,-

ST AKTIE
AKTIENVERWALTUNG
- Aktien- u. Depotverwaltung u. Bilanzierung
- CHART-Analyse incl. Datenbank mit Kurven mit über 50 versch. Aktien beliebig aktualisierbar u. erweiterbar
- Sämtl. großen deutschen Standardwerte (55 St.) bis Okt. 87 fortgeführt
- Kto.-Führung für alle Orders u. andere Buchungen
- Kompakt, übersichtlich, Menü-orientiert, einfache Bedienung durch Maus, u. lfd. Programmhilfen, Graph. Darstellung standardgemäß, generiert selbständig

DM 69,-

ST DIGITAL
LOGIKSIMULATOR FÜR DEN ATARI ST
Ein Programm zum Erstellen, Testen und Analysieren von Logikschaltungen für Ausbildung und Hobby-Elektronik.
- Komfortable GEM-Umgebung
- Bauteile lassen sich per Maus plazieren u. verdrahten
- Umfassendes Bauteile-Set (Grundgatter, Ein-/Ausgabe-Bausteine, Flip-Flops, etc.)
- Definition zusätzlicher Bauteile durch Makrotechnik
- Makros können in Libraries gespeichert werden
- Interaktive Simulation mit Darstellung der Leitungszustände (d. h. Betätigung von 'Schaltern' mit der Maus und sofortige Reaktion der Schaltung)
- Erzeugen von Impulsdiagrammen
- Hardcopy-Funktion
- Hardwarevoraussetzung: ST/Monochrom-Monitor

DM 89,-

SKYPLOT plus
EIN ASTRONOMIEPROGRAMM FÜR STERNENFREUNDE
- zwei Sterndatensätze mit 612 bzw. 15 383 Sternen
- 1053 Nebel, Sternhaufen u. Galaxien
- acht Planeten, Sonne, Mond u. drei Kometen vordefiniert
- Eingabe eigener Bahnelemente möglich
- Darstellung des Himmels als rechteckige Karte
- maßstäbliche Darstellung von Nebeln u. Planeten als Scheiben, letztere als Phrasen – etc. dadurch Darstellung von Finsternissen, Durchgängen, Bedeckungen etc.
- automat. Suche nach Konjunktionen
- Bewegungsbahnen von Planeten mit max. 32 000 Punkten, speicherabh.
- einstellbarer Datumsbereich 1.1.1583 – 31.12.3000
- Gradnetz auf Wunsch einblendbar
- für Hardcopy Bild invertierbar
- Darstellung dreidimensional oder von Planeten anderer Sterne aus
- Stellarstatistik; Sichtbarkeitsdiagramme
- Sternkarten können mit allen Objekten auf Plotter ausgegeben werden
- Simulation eines Films im Speicher, dadurch Bewegungsabläufe sichtbar
- Handbuchfunktion verweist bei jedem Menüpunkt auf Handbuchseite
- und vieles andere mehr
- Hardware-Voraussetzung: ATARI ST mit mind. 1 MByte.

DM 198,-
Update DM 80,-

ST-LEARN
DER VOKABELTRAINER FÜR DEN ST
programmiert v. D. Owerfeldt, Gewinner der GOLDENEN DISKETTE '87 für das beste Lernprogramm.
- Läuft vollständig unter GEM
- „Intelligente" Auswertung der Benutzereingaben
- Fehlerhäufigkeit einer Vokabel wird berücksichtigt
- Berücksichtigung mehrerer Bedeutungen eines Wortes
- Vielfältige Möglichkeiten des Lernens und der Abfrage
- Integriertes Lernspiel „HANGMAN"
- Spezielle Auswertung für unregelmäßige Verben (bei Eingabe von „to go" werden die anderen 2 Formen nachgefragt)
- Bei offensichtlicher Ähnlichkeit der Wörter wird wahlweise ein 2. Versuch zugelassen
- Trotz Einordnung der Vokabeln nach Lektionen oder Wissensgebieten ständig schneller Zugriff auf **alle** Vokabeln (nur durch Größe des Speichermediums begrenzt)
- Voller europäischer Zeichensatz (Zugriff durch die Maus unter GEM)
- Auch für Farbmonitor in mittlerer Auflösung
- Wörterbuchfunktion durchsucht alle Files eines Speichermediums nach einer Übersetzung ab
- Verbessertes Eingabeformular

DM 59,-

ST STRUKTUR-PAINTER
DAS SUPER-PROGRAMM zum kreativen Erstellen von Graphiken aller Art – ohne Malkenntnisse
- Einfache Bedienung, ansprechende Menüs
- Automatischer Malmodus, Muster in Millionen Variationen erstellt
- 2 Füllroutinen, 2 Blockspeicher
- WYSIWYG
- Ausschneiden u. Einkopieren von beliebigen Formen
- Freihand, Verschiedene Pinsel, Füllmuster als Pinsel, Malen mit Bildausschnitten
- Füllmuster definieren ohne Editor; eine ganz neue Methode
- Füllmuster-Bibliotheken; über 200 Füllmuster gleichzeitig verfügbar
- Vergrößern, Verkleinern; 3 Endprodukte zur Auswahl
- Spiegeln, Drehen (dreht in 2 Richt. gleichzeit.)
- Stauchen
- Rahmen (1): Erzeugen von Rahmen aus allen Graphiken
  Rahmen (2): Automatische Rahmenberechnung
- Schatten: Automatische Schattenberechnung (3-D-Effekt)
- Teilmuster; Font's (23 Größen, 21 Arten, 4 Verknüpfungen
- Lineal: Einblendbare Einteilung
- Radierer u. Sprühdose beliebig einstellbar
- Druckertreiber für alle Epson- u. IBM-kompat. Drucker sowie Laserdrucker implementiert
- Schnelle Lupe mit: Punkt, Rechteck, Linie, Invertieren, Löschen
- Invertieren auf Knopfdruck während gezeichnet wird
- Voll Mausgesteuert
- Keine Kompatibilitätsprobl. mit Bildern anderer Programme
- STruktur-Painter Bilder auch mit Signum 2 verarbeitbar.
- und viele weitere Möglichkeiten
- Hardwarevoraussetzung: ST mit mind. 512 KB frei

Ein Spitzenprogramm **DM 89,-**

* alle Preise sind unverbindlich empfohlene Verkaufspreise

Bitte besuchen Sie uns in Halle 7 / Stand E 46



16. – 23. MÄRZ 1988

Heim Verlag

Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 0 61 51 - 5 60 57

BESTELL-COUPON
an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Bitte senden Sie mir: ____________________

____

zzgl. DM 5,- Versandkosten (unabhängig von bestellter Stückzahl)
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ____________________

Straße, Hausnr. ____________ PLZ, Ort ____________

Benutzen Sie auch die in ST COMPUTER vorhandene Bestellkarte.

AHDI Command Set

Das ATARI-Hardisk-Interface hat leider einen recht bescheidenen Befehls-Umfang. Es kommen lediglich sieben Befehle zur Anwendung, die aber zum Betrieb der Festplatte ausreichen.

Diese sind:

1. Test Unit Ready ($00)
Mit diesem Kommando kann die Betriebsbereitschaft des Laufwerkes abgefragt werden.

2. Request Sense ($03)
Hiermit wird ein spezieller Prüfblock abgerufen, der u.a. eine definierte Fehlermeldung enthalten kann.

3. Format Unit ($04)
Ähnlich einer Diskette muß die Festplatte formatiert werden. Die notwendigen Informationen für das Format müssen vorher über den Mode-Select-Befehl übermittelt worden sein, sonst werden voreingestellte Daten benutzt.

4. Read ($08)
Entspricht dem Read-Befehl beim Diskettenlaufwerk zum Lesen der Daten von der Festplatte.

5. Write ($0A)
Das Ganze umgekehrt. Wie beim Read-Befehl wird ab der angegebenen Blockadresse die in Byte 4 eingetragene Anzahl von Datenblöcken in diesem Fall geschrieben.

6. Seek ($0B)
Auf diesen Befehl hin wird sofort die angegebene Blockadresse angesteuert und mit einer Statusmeldung quittiert.

7. Mode Select ($15)
Ein wichtiger Befehl, mit dem mindestebns 12 Bytes an Informationen für die Formatierung der Festplatte übertragen werden.

troller!). Die Unterlagen sind aber anscheinend nie korrigiert worden, und so tauchen diese falschen Befehle durchaus auch in aktueller Literatur auf.
Wenn man sich die Befehle betrachtet und mit den Orginal-Befehlen des Controllers vergleicht, fällt im ersten Moment kein Unterschied auf. Aber (Tramiel, s.o.) auf den zweiten Blick -und überhaupt, wo sind denn die Befehle der Gruppen 2? Die Kommandostruktur beim SCSI sieht im Byte 0 nämlich ein klein wenig anders aus: (siehe Kasten vorhergehende Siete unten) Der SCSI-Befehlssatz kann insgesamt bis zu acht Kommandogruppen mit je 32 Kommandos umfassen. Da aber die Gruppennummer im gleichen Moment am SCSI-Bus anliegen müßte, wenn der ST sein Target addressiert, werden die Bits 5 bis 7 am SCSI-Bus einfach per Hardware im Hostadapter für diesen Moment auf Null gelegt. Damit sind aber ausschließlich Gruppe-0-Kommandos möglich.

Original und Fälschung

Aber es kommt noch schlimmer. Der vom Hostadapter erzeugte SCSI-Bus ist gar keiner. Wenn Sie sich an die128 Festplatten erinnern - sie waren nur durch die Auswahl des Targets und des SCSI-Kanals möglich (bis zu acht Geräten am SCSI-Bus...) - denkste... Wieder spielt uns hier der Hostadapter einen Streich. Das Ablaufprotokoll eines SCSI-Busses sieht etwas umfangreicher aus:

1. Bus-Free-Phase
2. Selection-Phase
3. Commando-Phase
4. Data-In-/Out-Phase
5. Status-Phase
6. Message-Phase

(Es gibt noch ein paar mehr, die aber hier nicht so wichtig sind). Wie oben bereits erklärt, überwacht der ATARI den Bus selber; Punkt 1 fällt also flach und damit eben leider auch die Möglichkeit, Geräte direkt miteinander kommunizieren zu lassen. Punkt 2 findet statt (!), und zwar direkt im Hostadapter. Normalerweise wird zur Selektion eine der acht Datenleitungen gesetzt (das ist die “Hausnummer” für das Gerät am SCSI-Bus) und durch Setzten der Select-Leitung eine Antwortmeldung über die Busy-Leitung vom Target ausgelöst. Damit hat aber auch dann das Target das Sagen im Datenaustausch, denn erst nach Löschen der Busy-Leitung durch das Target können andere Teilnehmer wieder auf den Bus zugreifen!

Geschummelt

In unserem Fall wird einfach nach einem Reset (vom ST - also auch nach dem Einschalten) kurz die Datenleitung 0 und die Select-Leitung am SCSI-Bus vom Hostadapter gesetzt. Der Controller in der SH-204/205 wird also nur ein einziges mal angewählt; darum muß auch die Anlaufphase und Selbstinitialisierung der Festplatte abgewartet werden, eine Kontaktaufnahme mit dem SCSI-Teilnehmer ist sonst nicht mehr möglich. Nichts war's mit unseren 128 Festplatten, wir kommen maximal auf 8 Targets am ACSI-Bus x 1 Teilnehmer am SCSI-Bus x 2 Festplatten pro Controller = 16 Stück - wenn die Software jetzt noch mithalten könnte.

Frust und Freud

Eine gute und eine schlechte Nachricht zum Schluß - die gute zuerst: in der SH-205 ist der Fehler behoben. Jetzt die schlechte: anscheinend wird durch ein falsches Hardware-Timing im Hostadapter der SH-204 die Datenleitung in dem gleichen Moment auf Null gelegt, in dem eigentlich die Statusmeldung an den ATARI gehen müßte. Jede Art von Fehlermeldung wird also einfach ignoriert. Wehe wenn...

Zum Schluß...

...zu diesem Teil der Festplattenserie muß ich zum “Zum Schluß...” vom letzten mal eine Korrektur vornehmen, ich habe die Kurve einfach nicht gekriegt. Die dort versprochenen Ausführungen zum Thema Formatieren und Umbau mußte ich aus Platzmangel auf die nächste Folge verschieben...

hp

ENDE

CHACUN À SON GOUT...

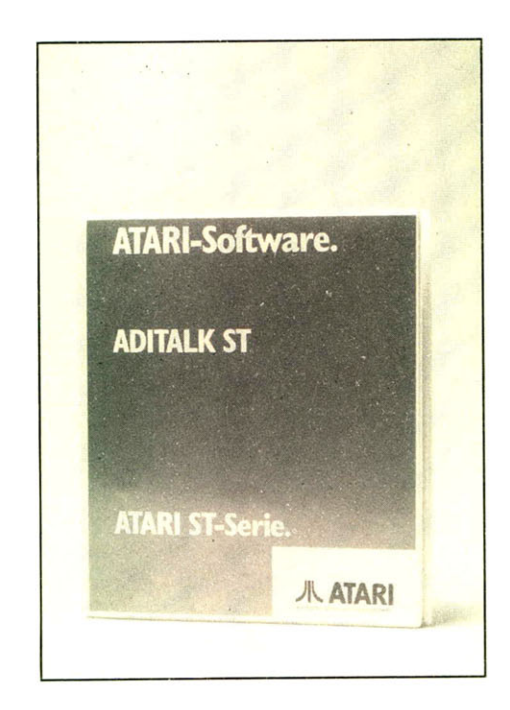
Programmiersprache ADITALK

Nachdem Anfang November Adimens 2.1 erschien, folgte Mitte des gleichen Monats die schon lange angekündigte Programmiersprache namens Aditalk. Aditalk muß nicht unbedingt zusammen mit Adimens 2.1 betrieben werden, es ist auch als eigenständiges Programm funktionsfähig, da es alle notwendigen Datenbankfunktionen zur Verfügung stellt und alle notwendigen Module mitgeliefert werden.

Neben Aditalk vertreibt ATARI auch noch die Programme C-PROG und BAS-PROG von ADI, die beide allerdings um die 2000 Mark kosten und sich mehr an den professionellen Anwendungsautor wenden.Mit diesen Programmbibliotheken ist es möglich, alle Datenbankfunktionen von Adimens in selbstgeschriebenen C- oder BASIC-Programmen zu verwenden.

Auch für Anfänger geeignet

Aditalk ist mehr für den Anwender gedacht, der einfache Prozesse automatisieren oder vehältnismäßig simple Aufgaben lösen will.
Das Programm wird mit Handbuch auf einer doppelseitigen Diskette geliefert. Neben der Programmiersprache und der Datei für die Hilfe-Funktion befinden sich



auf der Diskette die schon von Adimens 2.1 bekannten Programme INIT, REORG und das jetzt MSH genannte COMMAND sowie eine einfache Auftragsbearbeitung als Beispiel, inklusive Datenbank und mehrerer Programmdateien.
Zumindest das Programm INIT hat seit dem Erscheinen schon mehrfache Überarbeitungen und Korrekturen erfahren, was auch in einer Datei auf der Diskette dokumentiert ist. Leider war nicht zu erfahren, wie registrierte Anwender in den Genuß dieser und zukünftiger Updates gelangen sollen, wird doch im Dementi des Handbuchs ausdrücklich sämtliche Verantwortung (und Kosten) für Reparaturen auf den Kunden abgewälzt. Es besteht aber Hoffnung, daß die Wirklichkeit nicht ganz so unfreundlich aussieht. Zumindest gibt es in der ATARI-Mailbox in Raunheim (06142-42262)eine eigene Ecke für die Anwender von Adimens und Aditalk.

und nicht nur dBASE-Kenner

Im Gegensatz zu landläufigen Gerüchten ist Aditalk nicht dBase-kompatibel, weder dBase II noch III. Man kann also nicht direkt Anwendungen übernehmen, die man etwa für dBMAN oder ein anderes kompatibles Programm entworfen hat. Allerdings hat man bei vielen Kommandos eine dBase sehr verwandte Syntax verwendet. Die Tücke bei der Übersetzung steckt aber im Detail, da manche Funktionen, die man liebgewonnen hat, nicht existieren bzw. vollkommen anders implementiert wurden. Wie auch bei EXEC (Adimens-Bedienoberfläche) sind die Funktionen zur Erstellung einer Datei ausgelagert, darum wird auch hier das Modul INIT mitgeliefert. Durch diese Trennung von Datenbearbeitung und -erstellung sind die Daten zwischen TALK, EXEC und allen anderen Modulen austauschbar.
Ich werde mich in diesem Test ausschließlich auf das Modul TALK beschränken, da alle anderen Module schon im Zusammenhang mit Adimens 2.1 bzw 1.6 besprochen wurden. Im Gegensatz zu EXEC und INIT ist TALK kein GEM-Programm sondern eine TOS-Anwendung, man muß also auf die Maus verzichten. TALK kennt im wesentlichen zwei Bedienmodi und zwei Datenbankzustände. Entweder befindet man sich im Dialogmodus oder es wird ein Programm abgearbeitet. Viele Funktionen stehen nur zur Verfügung, wenn eine Datenbank geöffnet ist. Im Dialogmodus sind die meisten Funktions-tasten belegt, was in einer Statuszeile angezeigt wird und davon abhängig ist, ob eine Datenbank geöffnet wurde oder nicht, und ob ein Indexschlüssel definiert wurde.
TALK kennt ebenso wie dBase eine Reihe von 'Schaltern', womit man verschiedene

Betriebszustände definieren kann. Zusätzlich zum normalen Befehl <SET...TO...> ist es möglich, per Funktionstaste ein Menü aufzurufen, in dem man diese Schalter verstellen sowie für jedes mögliche Ausgabegerät (Device) eine Umlenkung der auszugebenden Daten definieren kann. Gibt man etwa bei dem Device *Output* 'PRN:' als Ausgabedatei an, wird die Ausgabe zusätzlich zum Bildschirm auch über die parallele Schnittstelle an einen angeschlossenen Drucker ausgegeben, sofern man den Schalter *Output* auf ON stellt.

Neben den bereits von Adimens bekannten Datentypen CHAR (Zeichenkette), INTEGER (Ganzzahl), REAL (Fließkommazahl), PRICE (Geldbetrag) und DATE (Datum) gibt es noch den neuen Typ LOGIC (wahr/falsch), der von einer Reihe von Funktionen zurückgegeben wird. TALK verfügt über alle gängigen Operationen (- + * / NOT OR AND < <= > >= <> =) für Zahlenwerte und läßt bis zu 255 global gültige Variablen zu. Eine in einem beliebigen Teilprogramm definierte Variable ist also auch in jedem anderen Teil verfügbar. Lokale Variablen sind leider überhaupt nicht vorgesehen. Immerhin kann man aber einen Variablennamen mit einer neuen Variablen besetzen, falls die ursprüngliche Variable nicht mehr benötigt wird, oder solche Variablen mit <RELEASE> wieder löschen. Merkmals- und Variablennamen dürfen Buchstaben, Zahlen und den Unterstrich enthalten, Merkmalsnamen zusätzlich auch eine Reihe von Sonderzeichen, damit die vollständige Kompatibilität zu Adimens (EXEC) gewahrt bleibt. Solche Namen müssen allerdings mit 2 Doppelkreuzen (#) eingefaßt werden.

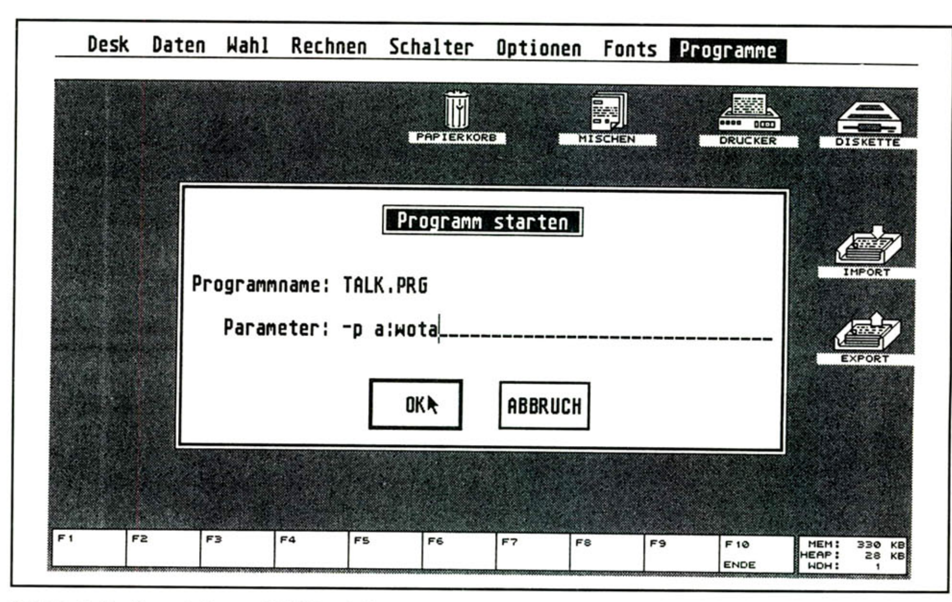


Bild 1: Aufruf aus Adimaus-EXEC mit Parameterübergabe

Zur einfachen Bearbeitung von Menüs mit mehreren Wahlmöglichkeiten steht neben den bereits von BASIC bekannten Verzweigungsbefehlen<IF..THEN..ELSE.. ENDIF> auch die Anweisung <DO CASE..CASE.. OTHERWISE.. ENDCASE> zur Verfü-gung. Auch gibt es im wesentlichen zwei Schleifenkonstrukte, nämlich <DO WHILE..ENDDO> für allgemeine Aufgaben und <ROLL.. ENDROLL> im Zusammenhang mit der Bearbeitung mehrerer Datensätze in einer geöffneten Datenbank. Die ROLL-Anweisung kann noch durch eine zusätzliche Bedingung auf bestimmte Datensätze beschränkt werden.

Als Ergänzung zu den Befehlen zur Datenmanipulation stehen einige Befehle zur Verfügung, die allgemeine Aufgaben übernehmen sollen. So gibt es einen <DEBUG>-Befehl, der einem Programmmierer bei der Fehlersuche behilflich sein soll. Doch leider passiert es oft genug, daß sich eine kleine Ungenauigkeit im Programm erst an sehr viel späterer Stelle stark auswirkt. Da hilft dieses ansonsten nützliche Utility leider nicht viel. Mit dem <RUN>-Befehl kann man ein externes Shell-Programm aufrufen, sofern genug Speicher zur Verfügung steht. Mit dem Befehlspaar <SUSPEND>,<RESUME> ist es möglich während der Programmentwicklung den Lauf z.B. zur Kontrolle von Variableninhalten (<SHOW MEMORY>) abzubrechen, und falls ein Programm lange läuft, kann man mit <BELL> den Benutzer auch akustisch auffordern, wieder an seinen ATARI zurückzukehren.

Es gibt zwei Möglichkeiten, ein Programm endgültig zu verlassen, mit <RETURN> kehrt man immer um eine Programmebene bis zum Dialogmodus zurück, mit <QUIT> verläßt man Aditalk. Damit man konstante Werte, die sich nur selten ändern, nicht im Programm oder in jedem Datensatz verewigen muß, gibt es das Befehlspaar <SAVE>,<RESTORE>, mit dessen Hilfe man ausgewählte Variablen in einer Datei ablegen kann und sie später bei Bedarf wieder zur Verfügung hat. Zum Beispiel kann man so den Mehrwertsteuersatz ablegen oder, falls man mehr als die zugelassenen 255 Variablen benötigt, einen Teil zwischenspeichern. Mit CLEAR MEM wird der gesamte Variablenspeicher gelöscht.

Aditalk arbeitet nach dem Prinzip eines halbkompilierenden Interpreters. Aus dem mit einem beliebigen externen Editor erstellten Sourcecode (Endung .TLK), in dem keine Textsteuerzeichen enthalten sein dürfen, wie sie etwa von 1st-word (plus) im WP-Modus produziert werden, wird in einem Übersetzungslauf der

```
WERBUNG.KUNDEN             kein Index               Dialog
==========================================================================
                                                    Kunde_Nr: 0000
  Firma:
  Hansi Discogeher..............
  Entertainment GmbH & Co KG....          Kürzel: Phreak____

  Name: Hansi_______________
  Vorname: Discogeher.....
  Anrede: r Herr
  Straße_Nr: Römerstr. 80.............
  PLZ: 76__  Ort: ....................
  Telefon: ...............

  Rechnung: .........
  Mahnung: .........  .........
  Sperrbetrag: ..........


  Letzter_Auftrag: 01.01.88         Gesamtumsatz: ..............
==========================================================================

 edit
 edit
1 INFO 2      3      4      5      6      7      8      9      10 STOP
```

Bild 2: Die Aufteilung des Bildschirms; hier bei Edierung einer Maske. In der ersten Zeile stehen die Angaben zur Herkunft der aktuellen Daten, in der letzten Zeile die momentane Belegung der Funktionstasten

Jetzt bringt der ATARI ST Ihr Büro in Form

ST-Kontor
Kaufmännische Anwendungen für alle ATARI ST-Computer
ST-Kontor
TOS-Manager
Kundenverwaltung
Lager + Fakturierung
Finanzbuchhaltung
Lohn + Gehalt
SYBEX
Finanzbuchhaltung
Software + Handbuch

Wußten Sie eigentlich, wieviel Arbeit Ihnen Ihr ST bei den täglichen Büro-Aufgaben abnehmen kann? Immer vorausgesetzt, daß Sie über eine Software verfügen, die alle nötigen Funktionen besitzt, komfortabel und schnell ist; wie die von der GFE R. Becker KG entwickelten Programme der Reihe SYBEX ST-Kontor (jeweils mit Trainingsbuch). Ihr ATARI ST (260–1040, MegaST 2–4) sollte einen Speicher von 1 MB RAM (für TOS-Manager und Kundenverwaltung reichen auch 512 KB) und das Betriebssystem TOS im ROM haben. Die mächtige FiBu benötigt ein double-sided Laufwerk mit 1 MB, optimal wäre eine Festplatte.

Natürlich finden Sie bei SYBEX noch mehr Software und Bücher zu Ihrem Rechner. Fordern Sie einfach unseren ATARI ST-Prospekt oder unser Gesamtverzeichnis an.

o hilft Ihnen der ST-Kontor TOS-Manager:

ist die neue Verwaltungszentrale Ihres ATARI-Rechners. Von hier aus können Sie alle übrigen ST-Kontor-Programme ohne nges Suchen aufrufen. Viele Accessories erleichtern Ihnen die Arbeit: RAMDisk, Taschenrechner, Terminkalender, ASCII-belle, Notizblock usw. Die Diskettenverwaltung ersetzt die GEM-Benutzeroberfläche und bietet ihnen viele zusätzliche Mög-chkeiten, wie: Formatieren und Kopieren von Disketten, Bearbeiten von Ordnern und Dateien, Suchen von Dateien, estaurieren gelöschter Dateien, Druckereinstellung. Ein Programm, mit dem Sie Ihre tägliche Arbeit einfacher und effizienter estalten.

est.-Nr. 3428, DM 98,–*/sFr. 98,–/S 872,–

Und das kann die ST-Kontor Kundenverwaltung:

Als Basismodul der ST-Kontor-Reihe ist die Kundenverwaltung für die Verwaltung Ihrer Adressen-Bestände zuständig; nur Ihre Festplatte bzw. das RAM können dabei Grenzen setzen. Die wichtigsten Features im Überblick:

- Superschnelles Suchen (Suchen nach Von-Bis-Werten in max. 0,5 Sekunden)
- Optimierte Datenspeicherung
- Mehrere Datenfelder pro Adresse (Stammdaten, Zusatzdaten, Notizen u. a.)
- Umfangreiche Selektionsmöglichkeiten nach beliebig vielen Kriterien
- Ausgabe auf Datei/Monitor/Drucker als Liste, auf Etikett, als Serienbrief

Best.-Nr. 3429, DM 149,–*/sFr. 149,–/S 1326,–

Ein Bündel arbeitssparender Features enthält die ST-Kontor Lagerverwaltung & Fakturierung:

Mit diesem starken Kombi-Programm bringen Sie Ihr Sortiment auf Vordermann: Stücklisten, Etiketten, Preislisten, Kataloge und umfangreiche Angebote mit Textbausteinen sind jetzt kein Problem mehr für Sie. Ebenso selbstverständlich sind verschiedene Lagerbestands-Bewertungsmethoden, Übersicht über den aktuellen Lagerbestand mit Bestellvorschlägen, Fakturierungsüber-sicht mit Offenen Posten, Teilen und Zusammenlegen von Fakturierungen – und die Übernahme in die Finanzbuchhaltung. Nicht zu vergessen die Bildschirmkasse mit dem aktuellen Kassenstand, Anzeige des Wechselgeldes und Ausdruck eines Kaufbelegs. Und betriebswirtschaftliche Funktionen zur Betriebsoptimierung werden direkt mitgeliefert.

Best.-Nr. 3430, DM 398,–*/sFr. 398,–/S 3542,–



Den aktuellen EG-Richtlinien entspricht die ST-Kontor Finanzbuchhaltung:

Die mehrfirmen- und mandantenfähige FiBu verdaut große Buchungsmengen mit einer erstaunlichen Geschwindigkeit. Unter anderem können Sie von ihr erwarten:

- Kontenrahmen SKR 04 mit 6000 Konten + Einzelhandels-Kontenrahmen SKR 02
- Verschiedene Bilanzierungs- und Abschlußmöglichkeiten mit einfacher Kostenstellen-Rechnung, Anlagenspiegel, Abschreibungs-Verzeichnis und betriebswirtschaftlicher Auswertung
- Verbuchen von Sammelbelegen mit maximal 10 Soll- und 10 Habenkonten sowie 4 Mehrwertsteuer-Sätzen pro Buchung
- Abwicklung des Zahlungsverkehrs mit Offenen Posten und Mahnwesen
- Übernahmedatei für Fakturierung/Lohn & Gehalt mit Korrekturmöglichkeiten

Best.-Nr. 3431, DM 498,–*/sFr. 498,–/S 4432,–

Das erledigt ST-Kontor Lohn & Gehalt für Sie:

In Vorbereitung

- Verwaltung der Mitarbeiterdaten samt Ausfüllen aller Versicherungs- und Finanzamtsformulare
- Alle Lohn- und Gehaltskonten auf einen Blick
- Verwaltung von Zeitkonten einschließlich Krankheits- und Urlaubstagen
- Abrechnung von Vermögensbildung, Direktversicherung und Lohnpfändung
- Komplette Erstellung der Lohnsteuer-Jahresdaten ohne Rechenaufwand
- Abwickeln des Zahlungsverkehrs mit Mitarbeitern, Ämtern und Versicherungen inklusive Berechnung der Zahlungsbeträge und Vorbereitung der Buchung. Die Daten können von der ST-Kontor FiBu übernommen und weiter bearbeitet werden.

Best.-Nr. 3432, DM 198,–*/sFr. 198,–/S 1760,–

*Unverbindliche Preisempfehlung

Übrigens: SYBEX sucht ständig gute Buch- und Software-Autoren. Interessiert? Dann kontakten Sie bitte Ralf Lieder, Tel. 0211/6180220.



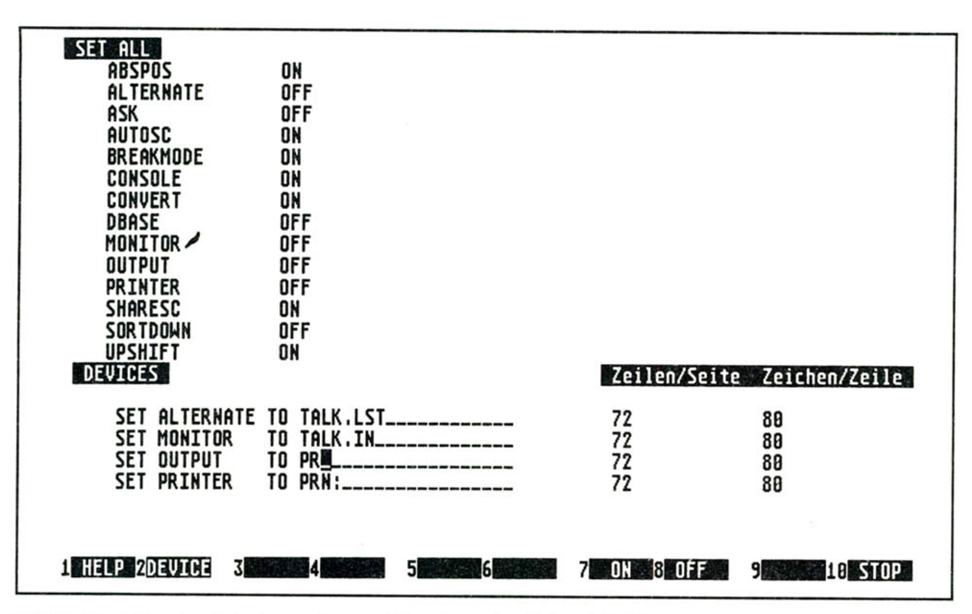


Bild 3: Einstellen der Schalter und sekundären Ausgabepfade mit SET ALL

ausführbare Zwischenkode (Endung .TLP) erzeugt. Im Zuge dieser Umwandlung werden die Befehlsworte in sogenannte Token umgewandelt und feste Werte vorberechnet. Man kann dieses Verfahren auch bei älteren Compilern finden, etwa beim UCSD-p-Pascal. Bei dieser Übersetzung gehen die Informationen über die Zeile, in der ein Befehl stand natürlich verloren, darum wird bei jedem auftretenden Fehler die Übersetzung mit einer entsprechenden Meldung abgebrochen, und der Programmierer kann den Fehler ausmerzen. Durch dieses Verfahren wird garantiert, daß die Programme bei der Ausführung zumindest semantisch fehlerfrei sind. Die übersetzten Programme werden mit dem Befehl <DO> aufgerufen. Ein einzelnes Programm kann 128 KByte lang sein, darf aber nur aus einer Ebene bestehen. Es ist jedoch möglich, mehrere Programme zu einer Maximaltiefe von 92 Ebenen zu verschachteln. Bei der Umwandlung werden übrigens alle Befehle, Variablen- und Merkmalsnamen in Großbuchstaben umgewandelt, schreibt man also in einem Programm

```
STORE 5 TO A
STORE 3 TO a
SUSPEND,
```

so findet man beim späteren Programmlauf beim Betrachten der Variablen folgendes Ergebnis:

```
A = 3
```

In diesem Zusammenhang sei gleich noch auf einen groben Fehler aufmerksam gemacht. Folgende Zeilen:

```
SET UPSHIFT OFF
/* Einschalten der */
/*Unterscheidung Groß/klein */
STORE "abc" to A
STORE "ABC" to B
? (A=B)
```

bringen nicht das erwartete Ergebnis 'FALSE' sondern die falsche Aussage 'TRUE'. Im Gegensatz dazu erhält man das korrekte Ergebnis, wenn man nur die jeweils ersten Zeichen mit Hilfe folgender Zeile vergleicht

```
? ASC(A) = ASC(B)
/* ASC ermittelt den ASCII-CODE */
/* des 1. Zeichens eines Strings */.
```

Wie bereits erwähnt, braucht man zum Erstellen eines Programmes einen externen Editor, da nicht alle Befehle im Dialogmodus ausgeführt werden können, insbesondere gilt dies für die Schleifen- und Bedingungs-Konstrukte. Es ist jedoch möglich, im Dialogmodus mehrere Zeilen zu 'Mikro'programmen zusammenzufassen, indem man an ein Zeilenende ein Semikolon anhängt. Alle so eingegebenen Zeilen werden en bloc ausgeführt, sobald eine Zeile ohne Semikolon eingegeben wird.

Mehr als 80 Kommandos

Aditalk unterscheidet zwischen 3 Kommandotypen: den Funktionen zum Arbeiten mit den Datenbeständen, den allgemeinen Befehlen und den allgemeinen Funktionen.

Zur ersten Gruppe gehören Befehle wie <OPEN> und <CLOSE> zum Öffnen bzw. Schließen einer Datenbank sowie <USE> und <INDEX>, mit deren Hilfe man die zu bearbeitende Datei und, falls gewünscht, den Suchschlüssel bestimmt. Mit <FIRST>, <LAST> und <SKIP> kann man sich in der aktuellen Datei bewegen und mit <JUMP..TO..>, <SELECT> und <BACK> ist der Zugriff auf mehrere Dateien einer Datenbank gleichzeitig möglich.

Zur Suche in den Daten stehen die Befehle <FIND> und <LOCATE> zur Verfügung, wobei <FIND> nur nach Schlüsseln suchen kann und daher wesentlich schneller ist als sein Gegenstück. Auch der bereits erwähnte Befehl <JUMP..TO..> kann zur Suche eingesetzt werden, wobei der Schlüssel auch aus einer anderen Datei stammen kann. Beim Suchen mit <JUMP..TO..> wird der aktuelle Datensatz im Gegensatz zu den Bewegungs- und Suchbefehlen nicht überschrieben,

geht weiter...

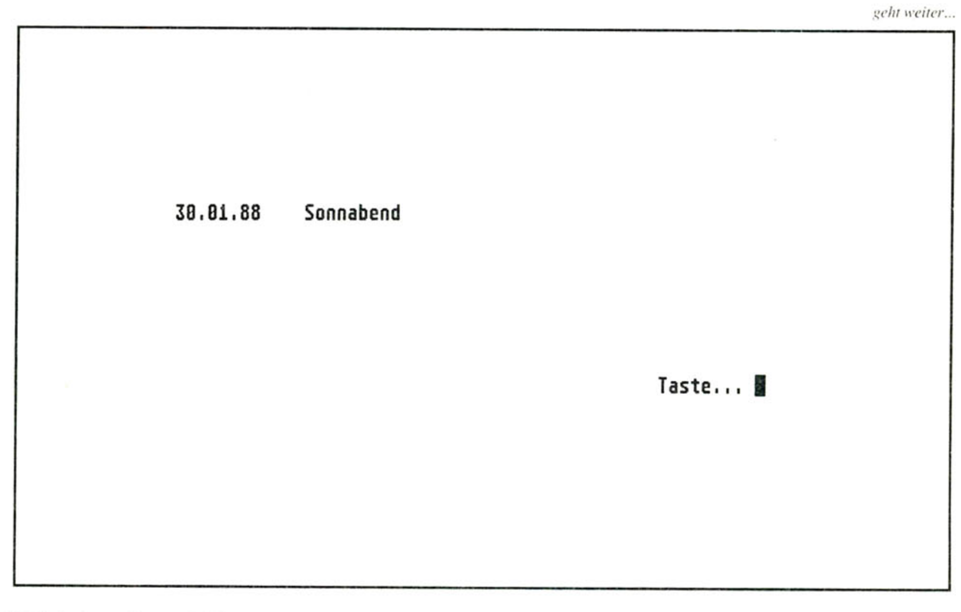


Bild 4: Ausgabe von WOTA:TLK

wodurch man bis zu 3 Datensätze gleichzeitig im Speicher halten kann.
Ein schönes Anschauungsbeispiel für den JUMP-Befehl wird in der mitgelieferten Beispielanwendung vorgeführt.
Ob man bei einer Suche eventuell über den letzten Datensatz hinausgeschossen ist, kann man mit der Funktion <EOF> (end of file, Dateiende) feststellen. Zur Manipulation der Datensätze stehen die Befehle <NEW> (Leeren des Datensatzes), <REPLACE..WITH..> und <EDIT> zur Verfügung. Wird ein Datensatz ediert, kann man sich mit der Taste F1 die Definition für das aktuelle Merkmal anzeigen lassen, also den Typ und die Länge. Allerdings werden erst durch <INSERT> (neu einfügen) oder <UPDATE> die Änderungen in die Datei geschrieben.
(Anm: Wird ein Datensatz mit <NEW> gelöscht, vergißt TALK auch, um welchen Datensatz es sich gehandelt hat. So kann man nicht mit <NEW> die Felder leeren, die Maske mit <EDIT> edieren und mit <UPDATE> zurückschreiben, wenn ein Datensatz komplett geändert werden muß, sondern muß alle Werte explizit auf Null, bzw. einen alternativen Ausgangswert setzen.)
Mit <DELETE> werden Datensätze in der Datei gelöscht. Es ist möglich, Datensätze ganz oder teilweise als Maske (<DISPLAY>) oder Liste anzuzeigen (<LIST>). Statistisch kann man die Daten mit den Funktionen <COUNT>, <MINIMUM>, <MAXIMUM>, <AVERAGE> und <SUM> auswerten.
Mit <SHOW STRUCTURE> kann man sich jederzeit die Definition der Datenbank anschauen.
Alle bisher genannten Befehle bezogen sich ausschließlich auf die in den Dateien abgelegten Datensätze und Merkmale, auch für Variablen sind die bereits weiter oben genannten Kontrollstrukturen sowie die nachfolgend aufgeführten Befehle gültig. So weist man einer Variablenliste mit der Anweisung <STORE..TO..> einen Wert zu, Ausdrücke wie

```
A = 5     oder
A = A + 1
```

sind leider nicht möglich, dafür aber z.B.

```
STORE 5 TO A,B,C ,
```

wodurch die Variablen A,B und C alle den Wert 5 erhalten.
Die Ein-/Ausgabebefehle sind äußerst eng an dBASE angelehnt, so bewirkt <?...> eine Ausgabe in eine neue Zeile, <??...> die Ausgabe ab der aktuellen Cursorposition und <@ X,Y ??...> die Ausgabe an der Position X,Y. Der Befehl <CLEAR> zum Löschen des Bildschirms kann ebenfalls mit einer X-Y-Koordinate angegeben werden, die dann der linken oberen Ecke des zu löschenden Quadranten entspricht.
Die Befehle <WAIT..TO..> und <ACCEPT..TO..> unterbrechen den Programmablauf und warten auf eine Eingabe von der Tastatur, die bei ACCEPT mit Return abgeschlossen werden muß.
Besonders hilfreich ist die Möglichkeit, die Funktionstasten auszuwerten, die die Zeichenketten "F01" bis "F10" erzeugen.
Die sogenannten Maskenbefehle sind allesamt von dBase bekannt.
Mit <SAY> werden Kommentartexte ausgegeben, mit <GET> können Merkmale und bereits definierte Variablen verändert werden. Die Eingaben bei <GET> können mit den Anweisungen <PICTURE> und <RANGE> eingeschränkt werden. Bis zu 64 GET-Anweisungen können parallel aufgerufen werden. Mit dem Befehl <READ> bzw. <READ SAVE> (wenn man die aktuelle Maske andernorts nochmal bearbeiten will) wird die Maske angezeigt und die Eingabe ermöglicht.

```
/* Beispiel Uhrzeit */
/* Die Anzeige sieht so aus: 12:00 */
CLEAR
STORE "12:" TO STUNDE
STORE "00" TO MINUTE
@ 3,5 SAY "Wie spät ? "
    GET STUNDE
    PICTURE "##^"
    RANGE "00", "23"
  @ ROW(),COL()+3 GET MINUTE
PICTURE "##"
     RANGE "00", "59"
READ
```

In diesem Beispiel kann man erkennen, daß jede logische Befehlseinheit in eine neue Zeile geschrieben werden kann, aber nicht muß. Eine Zeile darf aber nie länger als 80 Zeichen sein.
Die Funktionen von Aditalk zeichnen sich dadurch aus, daß ihre Parameter im Gegensatz zu den Befehlen in Klammern angegeben werden müsssen. Die erste von sechs Gruppen umfaßt alle Funktionen zur Behandlung von Zeichenketten (Strings). Mit ihnen kann man Strings vergleichen, in Groß- und Kleinbuchstaben umwandeln, Leerzeichenketten erzeugen, andere Typen in Strings umwandeln, Teilstrings ausfiltern, Leerzeichen aus Strings entfernen sowie von einzelnen Zeichen den ASCII-Code ermitteln bzw. aus dem ASCII-Code das Zeichen erzeugen.
Bei der Ausgabe von Strings muß darauf geachtet werden, daß man sie immer mit <+> verknüpft, da sonst besonders bei der Ausgabe von Steuerkodes für den Drucker oder den Bildschirm (VT-52 Emulation möglich) unerwünschte Ergebnisse auftreten können.
Mit der zweiten Gruppe der Funktionen lassen sich die logische Position des Cursors und des Druckkopfes feststellen.
Die dritte Gruppe umfaßt die Datumsfunktionen. Mit <DATE()> und <TIME()> lassen sich Systemdatum und -zeit feststellen. Es ist auch möglich aus dem Datum das Jahr, den Monat und den Tag zu extrahieren.
Außer den bereits genannten mathematischen Operationen stehen noch folgende Funktionen zur Verfügung, die in der vierten Gruppe zusammengefaßt sind: Exponentialfunktion, Logarithmus, Potenz und Quadratwurzel.
Zur fünften Gruppe gehören die <INT>egerfunktion, die aber nicht nur aus Fließkommazahlen ganze Zahlen macht sondern auch aus Strings, die nur aus Ziffern bestehen. Bei dBase macht das die hier nicht vorhandene VALuefunktion. Mit der <ROUND>-Funktion bestimmt man die Anzahl der signifikanten Stellen einer Zahl.
Die letzte 'Gruppe' besteht nur aus der Funktion <TYPE>, mit deren Hilfe man den Typ einer Variablen oder eines Merkmals bestimmen kann.

findet man im Handbuch

Mit diesen über 80 Kommandos, die noch durch diverse Parameter beeinflußt werden können, bietet Aditalk alles Nötige, um einfache und mäßig komplexe Anwendungen zu programmieren. Am Ende des Artikels findet sich ein Beispiel, mit dem es möglich ist, aus dem Datum den Wochentag zu bestimmen. Diese Funktion fehlt leider in TALK.
Das Handbuch zu Aditalk umfaßt knapp 200 Seiten, in denen die Module INIT, TALK und REORG beschrieben werden. Im Anhang sind die auch auf Diskette vorhandenen Sourcetexte für die Beispielanwendung nochmals abgedruckt.
Das TALK-Modul wird auf 125 Seiten beschrieben und enthält als einziger Teil neben dem Inhaltsverzeichnis auch einen Index, der sich allerdings aufgrund der Modularisierung mitten im Handbuch befindet. Wenn man schnell mal etwas

nachschlagen will, sollte man sich also Einlagen zwischen die Module legen.
Das Handbuch ist in zwei Teile gegliedert. Im ersten Teil werden die allgemeine Funktionsweise des Programms und die einzelnen Befehle und Funktionen nach Gruppenzusammengehörigkeit beschrieben. Im zweiten Teil, dem Referenzteil, werden jeweils die Befehle und Funktionen in alphabetischer Reihenfolge beschrieben. Die Trennung zwischen Befehlen und Funktionen ist hier aber eher hinderlich als nützlich, da man meist doch nur das Befehlswort kennt. Außerdem ist beim schnellen Durchblättern nicht genau erkenntlich, welche Kommandos auf einer Seite behandelt werden. Es wäre für den Anwender mit Sicherheit wesentlich einfacher, wenn auf jeder Seite nur ein Befehl abgehandelt würde. Oft werden im Handbuch auch Punkte an Stellen genannt, wo man sie nie vermuten würde, bzw. sie werden gar nicht erst aufgeführt. Es wird z.B. ein internationales Datumsformat erwähnt, jedoch wird nirgendwo erkärt, was es damit auf sich hat.
Dafür sind aber Stellen, die besonders wichtige Punkte beschreiben, entsprechend markiert.

Trotz mancher Fehler und Probleme

Im Verlauf des Tests habe ich mehrere kleine Anwendungen geschrieben und war mit TALK bis auf einige wenige Fehler sehr zufrieden, auf die ich größtenteils bereits an anderer Stelle hingewiesen habe. Ein letzter, ganz besonders seltsamer Fehler trat im Zusammenhang mit dem SAVE-Befehl auf. Im Dialogmodus funktionierte er einwandfrei, während er im Programmmodus bei der Übersetzung keine Variablennamen mehr annahm. Hier muß unbedingt Abhilfe geschaffen werden. Es gibt jedoch einige Punkte, die bei der Programmierung unbedingt beachtet werden müssen. So muß der Datensatzzeiger zu Beginn und Ende einer Programmschleife auf die gleiche Datei zeigen. Man muß insbesondere, wenn man mit <JUMP..TO..> in eine andere Datei verzweigt, auf der gleichen Schleifentiefe ein <BACK> durchführen, wenn man diesen Befehl in einem Schleifenkonstrukt benutzt. Ein anderer Punkt, bei dem man aufpassen muß, ist die Bildschirmausgabe. Wo immer möglich, sollte man explizit definieren, an welcher Stelle auf dem Bildschirm die Ausabe erscheinen soll, damit man keine Überraschungen erlebt. Es ist außerdem aufgefallen, daß das Setzen von Schaltern ein <CLEAR> auslöst. Wem es möglich ist, der kann ja mal die im Listing auskommentierte Zeile <SET OUTPUT OFF> wieder in die Ausführung einbeziehen.

empfehlenswert ?

Die Firma ADI hat mit Aditalk eine Daten bankprogrammiersprache auf den Markt gebracht, die alle Grundanforderungen und mehr erfüllt, die man an eine 'Query language' stellen darf. Aufgrund seines günstigen Preises wird es wohl in vielen kleinen Büros und vielleicht sogar bei der großen Gruppe der semiprofessionellen Anwender auf gute Resonanz stoßen, da sich gerade für diese Gruppe die Möglichkeit bietet, selbstgeschriebene Anwendungen zu vermarkten. Allerdings ist es wichtig, daß die genannten Fehler und Problemsituationen entfernt bzw. besser dokumentiert werden.

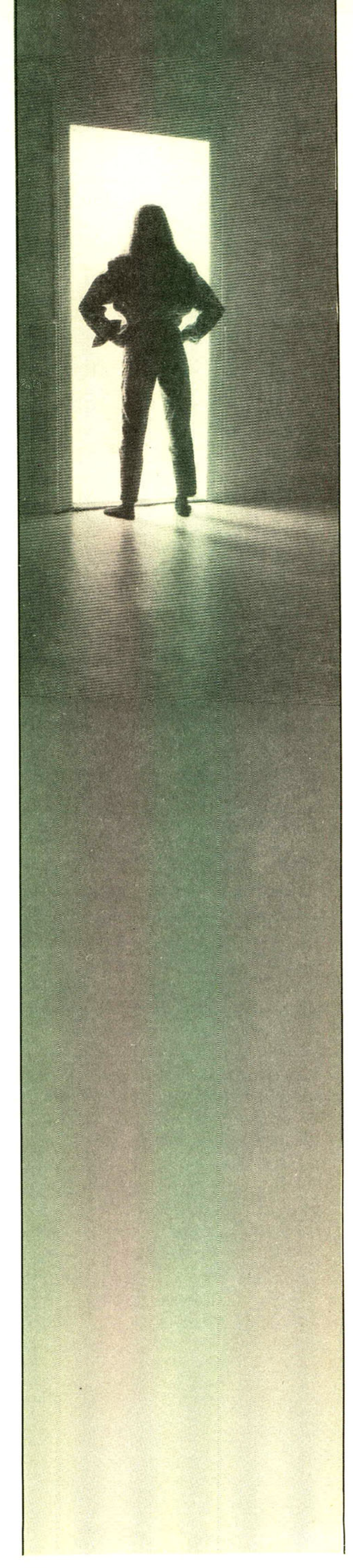
Christian Schmitz-Moormann

Bezugsadresse: Bei allen autorisierten ATARI-Händlern

```
/* Demonstrationsprogramm für Aditalk */
/* Name: WOTA.TLK */
/* Aufgabe: Berechnung eines Wochentags aus einem Datum
*/
/* Variablen : Vergleichsdatum, Tage, T, Tag */
/* by CSM */
CLEAR                         /* Bildschirm löschen */
SET SHARESC OFF               /* Keinen Platz für
                                 Masken reservieren */
SET OUTPUT TO "PRN:,64,80"    /* Parallele Ausgabe auf
                                 Bildschirm */
                              /* und Drucker
                                 vorbereiten und */
                              /* Drucker auf 64 Zeilen
                                 zu 80 Zeichen stellen
                                 */
STORE <25.12.99> TO Vergleichsdatum
@ 12,5 ACCEPT "Auch auf den Drucker ? (j/n)" TO Ok
IF Ok = "J" OR Ok = "F01"     /* Auch die F1-Taste als
                                 JA erkennen */
   SET OUTPUT ON              /* 2. Ausgabekanal
                                 (Drucker) öffnen  */
ELSE
   SET OUTPUT OFF             /* oder schlie_en */
ENDIF
@ 15, 5 ?? DATE()             /* Aktuelles Systemdatum
                                 3 Zeilen */
                              /* tiefer zeigen */
STORE Vergleichsdatum - DATE() TO Tage
STORE Tage - INT(Tage / 7.0) * 7 TO T
               /* INT (Tage / 7) * 7 liefert falsches */
                                          /* Ergebnis ! */
  DO CASE                     /* Wochentag bestimmen */
    CASE T = 0
      STORE "Sonnabend " TO Tag
    CASE T = 6
      STORE "Sonntag   " TO Tag
    CASE T = 5
      STORE "Montag    " TO Tag
    CASE T = 3
      STORE "Mittwoch  " TO Tag
    CASE T = 4
      STORE "Dienstag  " TO Tag
    CASE T = 1
      STORE "Freitag   " TO Tag
    CASE T = 2
      STORE "Donnerstag" TO Tag
    ENDCASE
@ row(),col()+3 ?? Tag      /* und hinter dem Datum
                               ausgeben */
/*SET OUTPUT OFF               Drucker aus       */
BELL                        /* Wecken */
@ 23,50 WAIT "Taste..."     /* und Bestätigung
                               abwarten */
QUIT                        /* oder RETURN, falls
                               TALK nicht */
                            /* verlassen werden
                               soll */
```

AUF DER SCHWELLE ZUM LICHT

Dateizugriff auf Massenspeicher



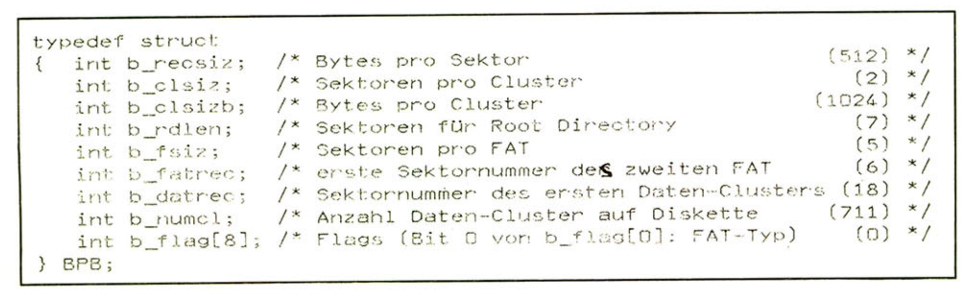
Heute möchte ich Ihnen etwas darüber erzählen, wie GEMDOS mit den Massenspeichern des ST (Diskette, Harddisk, RAM-Disk) umgeht, wenn es um das Lesen und Schreiben von Daten geht. Dafür ist die unterste Ebene der GEMDOS-Dateiverwaltung zuständig.

Als erstes wollen wir uns ein wenig um die Struktur der Speichermedien kümmern, auf denen sich bekanntlich neben den eigentlichen Dateien noch die Directories (Verzeichnisse) und die File Allocation Tables (FAT) befinden.
GEMDOS informiert sich über die Aufteilung eines Mediums mit der BIOS-Funktion 'Getbpb', die für jedes Laufwerk einen Zeiger auf einen 'BIOS Parameter Block' (BPB) liefert (Abb. 1). In Klammern sind die Werte für Disketten im Standard-Format angegeben. Die 'b_flags' werden bei Disketten vom BIOS für interne Zwecke benutzt. GEMDOS interessiert sich nur für 'b_flags[0]' (s.u.).
Bei Disketten werden die Daten für den BPB vom BIOS zwar aus dem Bootsektor gewonnen, doch GEMDOS interessiert sich überhaupt nicht für Bootsektoren, welche daher bei anderen Speichermedien auch überhaupt nicht erforderlich sind.

Intern werden FAT und Directories ganz ähnlich wie Dateien verwaltet, da so wesentliche Teile der Dateiverwaltung mitbenutzt werden können. Aus diesem Grund wird im folgenden meistens nur von 'Dateien' gesprochen werden; gemeint sind aber auch FAT und Directories.

Sektoren und Cluster

Ein Speichermedium ist in Sektoren, die die kleinste Einheit bilden, auf die GEMDOS Zugriff hat, unterteilt. Wie die Sektoren physikalisch auf z.B. der Diskette verteilt sind, soll ganz das Problem des BIOS bzw. der Laufwerks-Treiber bleiben, daher werden sie mit einer "logischen Sektornummer" angesprochen.



```
typedef struct
{  int b_recsiz;   /* Bytes pro Sektor                          (512) */
   int b_clsiz;    /* Sektoren pro Cluster                        (2) */
   int b_clsizb;   /* Bytes pro Cluster                        (1024) */
   int b_rdlen;    /* Sektoren für Root Directory                 (7) */
   int b_fsiz;     /* Sektoren pro FAT                            (5) */
   int b_fatrec;   /* erste Sektornummer des zweiten FAT          (6) */
   int b_datrec;   /* Sektornummer des ersten Daten-Clusters     (18) */
   int b_numcl;    /* Anzahl Daten-Cluster auf Diskette         (711) */
   int b_flag[8];  /* Flags (Bit 0 von b_flag[0]: FAT-Typ)        (0) */
} BPB;
```

Abb. 1: Struktur des BIOS Parameter-Blocks (BPB)

Für das Lesen und Schreiben von Sektoren ist die BIOS-Funktion 'Rwabs' zuständig. Hier sind alle Sektoren einfach von Null bis zu einem Maximalwert durchnumeriert. Diese Art der Numerierung wird von mir als 'BIOS-Zählung' bezeichnet.

Daten-Sektoren werden zu sogenannten Clustern zusammengefaßt. Ein Cluster besteht aus 1,2,4,.. (einer Potenz von 2) aufeinanderfolgenden Sektoren. Für jeden Cluster gibt es einen Eintrag in der FAT. Cluster werden von 2 an aufwärts durchnumeriert.

Viele Sektoren pro Cluster haben den Vorteil, daß die FAT wesentlich kleiner ist und daher schneller durchsucht werden kann. Außerdem arbeitet die unten erläuterte Pufferung von Sektoren bei wenigen FAT-Sektoren effektiver. Ein weiterer Vorteil ist die geringere "Zersplitterung" von Dateien, die sich darin äußert, daß eine Datei aus kleinen, auf das Medium verstreuten "Stücken", besteht.

Der Nachteil der Bildung von Clustern ist die größere Verschwendung von Speicherplatz, da der für eine Datei benötigte Platz immer auf ein Vielfaches der Cluster-Größe "aufgerundet" werden muß.

Im BPB steht die Sektornummer des ersten Daten-Sektors ('b_datrec').

File Allocation Table (FAT)

Über die FAT werden die Daten-Cluster den Dateien zugeordnet. Jeder Eintrag eines Clusters gibt an, ob er unbenutzt oder defekt ist oder gibt die Nummer des nächsten zur Datei gehörenden Clusters bzw. das Dateiende an. Im Directory ist der erste Cluster einer Datei vermerkt.

Jeder Eintrag der FAT ist 12 oder 16 Bit lang, je nach Speichermedium. Das Format wird durch Bit 0 von 'b_flags[0]' des BPB bestimmt (gesetztes Bit für 16-Bit-Format). Die genaue Struktur finden Sie in den 'Floppy-Spielereien' (ST 6/87, 1/88) ausführlich beschrieben (Gruß zurück an Claus!).

Die ersten zwei Einträge sind unbenutzt und werden von GEMDOS nicht beachtet (bei Standard-Disketten steht dort $F7FFFF). Dies ist wohl der Grund dafür, daß die Cluster-Numerierung bei 2 beginnt. Die Länge einer FAT steht im BPB unter 'b_fsiz'.

GEMDOS verwaltet zwei identische FATs, wie schon aus der Struktur des BPB ersichtlich wird. Beim Schreiben eines FAT-Sektors wird er stets in beide FATs geschrieben, die somit jederzeit identisch sind. Gelesen wird jedoch immer aus der zweiten FAT. Die erste FAT ist also nur eine Sicherheitskopie, auf die bei einer beschädigten zweiten FAT zurückgegriffen werden könnte. GEMDOS berücksichtigt dies allerdings nicht; man müßte sich also selbst ein Programm schreiben, das eine defekte FAT durch die Kopie ersetzt. Es ist leider nicht möglich, auf eine FAT zu verzichten, sie müssen sogar unmittelbar hintereinander liegen. Im BPB ist explizit nur der Beginn der zweiten FAT vermerkt ('b_fatrec'). Die erste FAT muß direkt davor liegen, also bei 'b_fatrec' minus 'b_fsiz'.

Directories

Das Root Directory (Hauptverzeichnis) genießt eine Sonderstellung unter den Directories, da es eine feste Größe und einen festgelegten Platz auf dem Speichermedium hat.

Es beginnt nach der zweiten FAT. Seine Länge ist im BPB festgelegt ('b_rdlen'). Aus der Größe eines Sektors (i.a. 512 Byte) und der eines Directory-Eintrags (32 Byte) ergibt sich die maximale Anzahl von Einträgen. Da alle Directory-Sektoren hintereinander liegen, braucht auf sie nicht mittels der FAT zugegriffen zu werden.

Subdirectories werden dagegen wie normale Dateien behandelt. Sie haben einen Eintrag im Parent Directory, ihre Cluster liegen im Datenbereich der Diskette und werden daher auch mittels der FAT verwaltet. Ihre Länge wächst mit der Zahl der Einträge und ist nur durch die Kapazität des Massenspeichers begrenzt. Ein Subdirectory wird jedoch nicht automatisch verkürzt, wenn Dateien gelöscht werden.

Wenn von "Daten" gesprochen wird, sind Subdirectories stets mit eingeschlossen; "Directory" meint nur das Root Directory. Den genauen Aufbau eines Directories können Sie in den 'Floppy-Spielereien' (ST 8/87) nachlesen.

Sektorzählung des GEMDOS

Um die Sache noch ein wenig zu komplizieren, zählt GEMDOS die Sektoren intern nicht wie das BIOS, sondern hat seine eigene Zählweise, in der die eben besprochene Strukturierung des Mediums zum Ausdruck kommt.

Daten-Sektoren haben in der GEMDOS-Zählung positive Sektornummern. Die Nummer des ersten Sektors des ersten Clusters (also von Cluster 2) ist zwei mal die Anzahl der Sektoren pro Cluster.

Die Sektoren des Root Directories und der FAT haben bei der GEMDOS-Zählung negative Nummern. Sie werden nach einem komplizierten Verfahren aus den Daten des BPB bestimmt, worauf wir erst nächsten Monat zurückkommen werden. Hier seien als Beispiel nur die Werte für Disketten im Standard-Format angegeben (Abb. 2). Wenn Sie es nicht abwarten wollen, können Sie ja versuchen, das "System" zu erraten (viel Spaß!).

Bei der Zählung der Datensektoren sind sich BIOS und GEMDOS allerdings nicht ganz einig. Der 'b_numcl'-Wert des BPB gibt die Gesamtzahl der vorhandenen Daten-Cluster an. GEMDOS ist der Meinung, dies sei die Nummer des letzten Daten-Clusters in seiner Zählweise. Da GEMDOS die Cluster von 2 an zählt, rechnet es mit zwei Clustern weniger, als eigentlich da sind. Dies ist durchweg bei allen internen GEMDOS-Routinen der Fall, so daß die letzten zwei Cluster eines Mediums von GEMDOS ungenutzt bleiben. Bei Disketten werden somit vier Sektoren (= 2 kB) Speicherplatz verschenkt.

	GEMDOS-Nr.	BIOS-Nr.
Bootsektor	----	0
FAT 1	----	1...5
FAT 2	-16...-12	6...10
Root Directory	-10...-4	11...17
Daten-Sektoren	4...1421	18...1435
unbenutzt	----	1436...1439

Abb. 2: Aufteilung einer Standard-Diskette

```
typedef BCB
{ BCB  *b_link;    /* Zeiger auf nächsten BCB dieser Liste */
  int   b_bufdrv;  /* Laufwerksnummer, -1 für ungültigen BCB */
  int   b_buftyp;  /* FAT (0), DIR (1), DATA (2) */
  int   b_bufrec;  /* Sektor-Nummer in GEMDOS-Zählung */
  int   b_dirty;   /* ungleich Null: Pufferinhalt geändert */
  DMD  *b_dmd;     /* Zeiger auf DMD von b_bufdrv */
  char *b_bufr;    /* Zeiger auf eigentlichen Sektor-Puffer */
} BCB;
```

Abb. 3: Struktur des Buffer-Control-Blocks (BCB)

Das Konzept der Sektor-Pufferung

Damit Sie auch wissen, warum ich Ihnen dies alles so genau erklärt habe, kommen wir nun zur Anwendung dieser Grundlagen, indem wir uns ansehen, wie GEMDOS nun eigentlich seine Zugriffe auf Massenspeicher abwickelt.

GEMDOS hat drei allgemeine Routinen zum Lesen bzw. Schreiben von Sektoren auf bzw. von Massenspeichern. Sie sind auch für die Übersetzung der GEMDOS-Zählung in die BIOS-Zählung (mit Hilfe des DMD) zuständig. Die übergeordneten Dateifunktionen kennen also nur GEMDOS-Sektornummern.

Es müssen einzelne Sektoren übertragen werden können, von denen nur einige Zeichen benötigt bzw. geändert werden sollen (vor allem bei FAT und Directories). Um zu verhindern, daß sie bei jedem Zugriff erneut geladen werden müssen, was eine ziemliche Zeitverschwendung wäre, ist es sinnvoll, solche Sektoren zwischenzuspeichern.

GEMDOS verwaltet hierfür zwei Pufferlisten, eine für FAT-Sektoren, die andere für Directory- und Daten-Sektoren. Mit Directory-Sektoren sind hier wieder nur die Sektoren des Root Directory gemeint.

Zu jedem Puffer existiert ein sogenannter "Buffer Control Block" (BCB). Jeder BCB enthält Angaben über den zugehörigen Sektor (Abb. 3), damit GEMDOS den Überblick behält (genauer gesagt: es versucht). Dazu gehören die logische Sektornummer in GEMDOS-Zählung, die Laufwerkskennung (0...15), der Puffer-Typ (0,1,2 für FAT-, Directory- bzw. Daten-Sektoren) und die Adresse des eigentlichen Puffers, wo der Inhalt des Sektors zu finden ist.

Eine Laufwerkskennung von -1 gibt an, daß der Puffer zur Zeit unbenutzt ist. Damit sind die anderen Daten des BCB bis auf 'b_bufr' ungültig.

Des weiteren gibt es ein "Dirty-Flag". Wenn es ungleich Null ist, wurde der Sektor geändert und ist noch nicht auf das Laufwerk zurückgeschrieben worden.

Zusätzlich zu seiner Kennung wird das Laufwerk noch durch den "Drive Media Descriptor" (DMD) identifiziert, bei dem ich Sie erneut auf die nächste Folge vertrösten muß.

Alle BCBs einer Pufferliste sind miteinander verkettet, d.h. 'b_link' zeigt auf den nächsten BCB, beim letzten BCB einer Liste ist 'b_link' gleich 0L (Abb. 4).

Die Anfänge der beiden Listen ist in der globalen Systemvariablen 'bufl' ($4B2) vermerkt: 'bufl[0]', also $4B2, enthält den Anfang der FAT-Liste, 'bufl[1]', also $4B6, zeigt auf den ersten BCB der DIR/DATA-Liste.

Die Sektoren sind in der Reihenfolge des letzten Zugriffs in der Liste sortiert. Der erste Sektor ist der zuletzt angesprochene, usw.

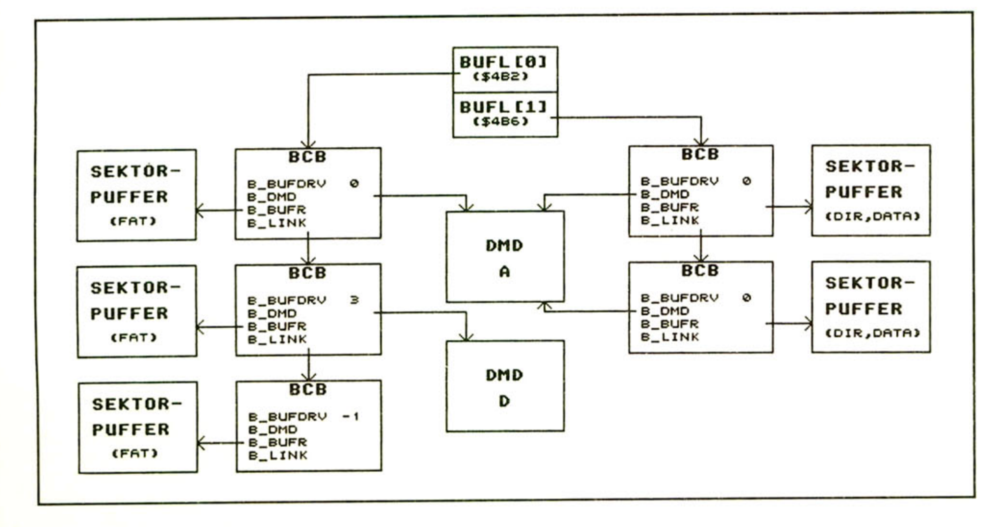


Abb. 4: Struktur der GEMDOS-Pufferlisten (Beispiel)

Sektor über Pufferliste übertragen

Zum Lesen eines einzelnen Sektors über die Pufferliste dient die im folgenden 'f_sread' genannte interne Routine. Sie findet auch bei Schreibzugriffen Verwendung, da auch dort der Sektor vor seiner Änderung erst einmal geladen werden muß.

Falls der zu lesende Sektor schon in der Pufferliste vorhanden ist, wird mittels der BIOS-Funktion 'Mediach' geprüft, ob das Speichermedium (i.a. die Diskette) gewechselt wurde. Bei einem "sicheren Mediumwechsel" wird die ganze GEMDOS-Funktion sofort mit der BIOS(!)-Fehlermeldung E_CHNG (-14) abgebrochen, wie in der Januar-Ausgabe beschrieben.

Tritt "nur" ein "möglicher Mediumwechsel" auf, so wird der Sektor einfach nochmal geladen, allerdings ohne Rücksicht darauf, ob er schon geändert wurde ("Dirty"-Flag)! Allerdings wird der "mögliche Mediumwechsel" vom BIOS nur (?) bei Disketten mit Schreibschutz gemeldet (dort allerdings fast immer!), so daß dieser Fall nicht eintreten sollte.

Wenn der Sektor noch nicht gepuffert ist, wird er in einen freien Puffer der jeweiligen Liste geladen. Ist keiner mehr unbenutzt, so wird der "älteste" Puffer (also der letzte) aus der Liste entfernt. Dadurch wird erreicht, daß die am häufigsten benötigten Sektoren am längsten gepuffert bleiben.

Dabei wird der Sektor natürlich zurückgeschrieben, falls er geändert wurde. Er wird mit 'BIOS-Rwabs' gelesen. Tritt hierbei ein Fehler auf, wird die GEMDOS-Funktion wie üblich abgebrochen.

Der zu lesende Sektor wird konsequenterweise in jedem Fall an die erste Stelle der Liste gehängt, egal, ob er tatsächlich geladen wurde oder schon vorhanden war.

Am Ende von 'f_sread' wird das 'Dirty-Flag' gesetzt, wenn ein Schreibzugriff geplant ist. Die übergeordneten Funktionen zum Schreiben rufen 'f_sread' kurz vorher auf, so daß diese Methode gerechtfertigt ist. Dieses Verfahren hat den Vorteil, daß nur die auf der untersten Ebene angesiedelten Funktionen für die Pufferliste sich mit den Details der BCBs herumschlagen müssen.

Nun sind auch noch ein paar Worte zur oben schon erwähnten Routine zum Schreiben einzelner Sektoren ('f_swrite') angebracht. Sie wird außer bei 'f_sread' immer dann gebraucht, wenn explizit

bestimmte Sektoren zurückgeschrieben werden sollen (z.B. beim Schließen einer Datei). Bei FAT-Sektoren finden die Schreibzugriffe für die beiden identischen FATs unmittelbar nacheinander statt.
Bemerkenswert ist, daß vor dem eigentlichen Scheibvorgang (mit Rwabs) der Puffer ungültig gemacht (b_bufdrv = -1), und erst danach wieder für gültig erklärt wird (außerdem wird 'b_dirty' natürlich gelöscht). Dies hat zur Folge, daß nach einem Abbruch der GEMDOS-Funktion bei einem Schreibfehler der Puffer ungültig ist, d.h. daß sein Inhalt als verloren angesehen wird.
In einigen Fällen wird 'f_swrite' auch mit einem ungültigen oder nicht veränderten Puffer aufgerufen. Diese werden zwar nicht geschrieben, wie es auch selbstverständlich sein sollte, aber veränderte werden ungültig gemacht. Dies hat weitreichende Folgen, wie Ihnen bald klar werden wird.

Direkter Sektor-Zugriff

Wenn alle Zugriffe nach dem oben beschriebenen Verfahren ablaufen würden, wäre das Laden von Programmen vermutlich genauso langsam wie das Lesen von Texten mit 1st Word+.
Daher gibt es eine weitere elementare Routine für den Massenspeicher-Zugriff, die für das Übertragen einer zusammenhängenden Folge von Sektoren zuständig ist ('f_mrw'). Sie arbeitet also wie 'Rwabs' auf BIOS-Ebene, hat ähnliche Parameter und ruft im Prinzip diese Funktion direkt auf.
Hier müssen allerdings Kollisionen mit der Pufferliste berücksichtigt werden. Aus diesem Grund werden alle Sektoren der Pufferlisten, die nun durch 'f_mrw' übertragen werden sollen, zuerst mit 'f_swrite' zurückgeschrieben. Dann erst erfolgt die Übertragung mit 'Rwabs'.
Beim Lesen wird dadurch sichergestellt, daß Änderungen in der Pufferliste nicht unter den Tisch fallen. Der Schreibzugriff könnte zwar vermieden werden, indem 'f_mrw' sich die geänderten Sektoren aus der Pufferliste holt, doch soviel Mühe wollten sich die GEMDOS-Programmierer mit 'f_mrw' offensichtlich nicht machen.
Beim Schreiben ist dies sogar ganz überflüssig, da der Sektor ja sowieso gleich ganz neu geschrieben wird. Hier wäre es allerdings wichtig, den entsprechenden Puffer ungültig zu machen (wenn er schon nicht auf den neuesten Stand gebracht wird), da nachfolgende Schreibzugriffe über die Pufferliste sonst auf den alten Sektorinhalt gehen. Wie wir gerade gesehen haben, wird dies von 'f_swrite' aber nur gemacht, wenn der Sektor nicht geändert wurde.

Und GEMDOS ist doch nicht fehlerfrei

Welche fatalen Folgen dies haben kann, sehen Sie an Listing 1. Ihre Kenntnisse in C oder einer ähnlichen Sprache sollten Ihnen sagen, daß nach Ablauf dieses Programms in der Datei 'Test' ein 'c' und 511 'b's stehen. Ein kurzer Blick in 'Test' nach dem Programmlauf wird Ihnen allerdings ein 'c' und 511 'a's bescheren! Die Erklärung dürfte nach den vorangegangenen Erläuterungen und den Kommentaren im Listing nicht schwerfallen.
Moral von der Geschichte: Benutzen Sie 'Fwrite' immer nur für Datenmengen größer oder kleiner als 512 Byte. Eine 'Mischung' ist allerdings erlaubt, wenn Sie rein sequentiell arbeiten, also kein 'Fseek' verwenden.
Hiermit dürfte klar geworden sein, daß die "Zusammenarbeit" zwischen den beiden Arten des internen Datenzugriffs nicht gerade besonders gut ist. Glücklicherweise treten diese Fehler in der Praxis oft nicht auf, da die Pufferliste hauptsächlich bei FAT und Directories in Erscheinung tritt, der Mehr-Sektor-Zugriff vornehmlich bei größeren Dateien.
Es gibt noch einen weiteren Fehler im Zusammenhang mit der Pufferliste. Unter bestimmten, nicht geklärten Umständen hat plötzlich ein Daten-Sektor der Pufferliste die GEMDOS-Sektornummer 0. Diese Sektornummer kommt ja normalerweise gar nicht vor, was beim Zurückschreiben aber nicht bemerkt wird. Bei 2 Sektoren pro Cluster hat der erste Daten-Sektor die GEMDOS-Nummer 4, also wird dieser "Sektor 0" auf den vierten Sektor vor den ersten Daten-Sektor geschrieben. Beim Standard-Diskettenformat ist dies der viertletzte von sieben Sektoren des Root Directorys (RD). Da das RD selten mehr als 48 Einträge (das sind drei Sektoren) hat, merkt man von diesem Fehler normalerweise nichts.
Bei der Programmierung eines RAM-Disk-Treibers war ich nun zufällig der Meinung, vier RD-Sektoren würden's auch tun. Daraufhin überschrieb mir GEMDOS regelmäßig meinen ersten RD-Sektor... und ich brauchte zwei Tage, um den Fehler GEMDOS und nicht dem RAM-Disk-Treiber zuzuschreiben. Auch auf einigen Disketten konnte ich mittels eines Diskettenmonitors einen vermurksten vierten RD-Sektor finden.
Bei mir trat dieser Fehler immer nur auf, wenn ein Programm Schreibzugriffe in "kleinen Einheiten" (also über die Pufferliste) machte und dabei das Speichermedium voll wurde. Aber vielleicht weiß jemand von Ihnen ja mehr darüber?

Datei-Zugriff

Nachdem wir nun die elementaren Zugriffsroutinen besprochen haben, und Sie hoffentlich noch interessiert dabei sind, geht es nun um den Zugriff auf Dateiebene. Dazu gibt es eine umfangreiche Routine(von mir 'f_frw' genannt), über die alle Dateizugriffe laufen. In der Programmhierarchie direkt darüber "sitzen" die GEMDOS-Funktionen 'Fread' und 'Fwrite', so daß Sie eine Vorstellung davon haben, was von 'f_frw' geleistet werden muß - nämlich die Umsetzung der relativen Datei-Positionen in die logischen Sektornummern (GEMDOS-Zählung), die von 'f_sread' & Co. verstanden werden. Es muß ferner möglich sein, von einer beliebigen Position innerhalb einer Datei eine beliebige Anzahl von Zeichen zu übertragen.
Die Parameter sind ähnlich 'Fread'/'Fwrite', statt des Datei-Handles wird ein interner 'File Descriptor', der auch für Directories usw. existiert, übergeben (dazu mehr in einer späteren Folge).
Aus der aktuellen Dateiposition und den Laufwerks-spezifischen Daten wie Cluster-Größe usw., die im DMD stehen, wird die logische Sektornummer und die Position des ersten Zeichens, auf das zugegriffen werden soll, errechnet.
Im allgemeinen wird der Zugriff mitten in einem Sektor beginnen. Falls dies der Fall ist, wird er mit 'f_sread' über die Pufferliste geladen. Der tatsächlich interessierende Teil der Daten wird dann vom bzw. zum vom Aufrufer bereitgestellten Speicherbereich kopiert. Falls alle zu übertragenden Bytes in diesem Sektor liegen, ist alles erledigt.
Ansonsten werden alle nun folgenden Sektoren, die komplett übertragen werden müssen (das sind also alle restlichen, eventuell bis auf den letzten), gelesen bzw. geschrieben. Hierbei entfällt der Umweg über die Pufferliste.
Zuerst werden die restlichen Sektoren bis zum Ende des aktuellen Clusters auf einen Schlag mit 'f_mrw' übertragen. Dies ist möglich, da alle Sektoren eines Clusters in der logischen Sektornumerierung aufeinander folgen.

Nun kommen die komplett zu lesenden Cluster dran. Dabei kann es natürlich vorkommen, daß die Cluster über das Medium verstreut sind. GEMDOS ist nun so schlau, die Cluster in möglichst großen Gruppen zu übertragen, was die Geschwindigkeit erhöht. Es geht die Cluster-Nummern, die es aus der FAT holt, durch, bis es eine "Lücke" entdeckt. Dann wird die auf diese Weise ermittelte zusammenhängende Cluster-Gruppe in einem Rutsch übertragen.

Beim letzten Cluster wird es im allgemeinen so sein, daß nicht alle Sektoren gebraucht werden. Diese werden separat behandelt, genauso wie die letzten Sektoren des ersten "angebrochenen" Clusters. Auch hierbei wird natürlich direkt über 'f_mrw' gearbeitet.

Zu guter Letzt bleibt unter Umständen ein Sektor übrig, bei dem nur auf den ersten Teil zugegriffen werden soll. Dies geschieht wie beim ersten Sektor über die Pufferliste und anschließendes Kopieren.

Zurückgegeben wird die Anzahl der tatsächlich übertragenen Bytes. Dies wird von 'Fread' und 'Fwrite' direkt an den Aufrufer zurückgegeben. Bei voller Diskette oder Dateiende wird einfach abgebrochen, so daß dies beim Vergleich des Rückgabewertes mit der gewünschten Zahl der zu schreibenden Zeichen bemerkt werden kann.

Sie sehen also, daß GEMDOS hier recht wirkungsvoll arbeitet, insbesondere, wenn große Datenmengen auf einmal übertragen werden.

Bei der Übertragung ganzer Cluster wird die FAT gelesen, um aus der Dateiposition die Clusternummer zu bestimmen. Da ja auch die FAT intern wie eine Datei verwaltet wird, geschieht dies mit einer Routine, die letztendlich wiederum 'f_frw' aufruft. Da bei FAT-Zugriffen aber immer nur einzelne Bytes gelesen werden, bricht diese Rekursion immer hier schon ab.

Das Lesen von ganzen Dateien mit nur einem 'Fread' ist daher sehr schnell; eine Verzögerung tritt dann auf, wenn ein neuer FAT-Sektor, der nicht in der Pufferliste vorhanden ist, geladen werden muß.

Programme, die Dateien in sehr kleinen Portionen, im Extremfall byte-weise, lesen, sind nicht langsam, weil die Sektoren zu oft geladen werden (dies wird durch die Pufferliste ja verhindert), sondern weil das ganze Drumherum einfach zu lange dauert. Bis das Programm das letzte Byte verarbeitet, den nächsten Aufruf von 'Fread' gemacht und GEMDOS sich bis zu der Stelle vorgekämpft hat, wo der Sektor wirklich geladen wird, ist so viel Zeit vergangen, daß das BIOS den nächsten Sektor gerade verpaßt hat (die Diskette dreht sich bekanntlich immer weiter). Und bis der Sektor sich mal wieder unter dem Schreib-Lese-Kopf vorbeibewegt, dauert es schon eine Weile.

Anwenderprogramme haben normalerweise die Möglichkeit, die Dateizugriffe in großen Einheiten durchzuführen, GEMDOS selbst muß jedoch bei FAT- und Directory-Operationen immer auf einzelne FAT- bzw. Directory-Einträge (2 bzw. 32 Bytes) zugreifen.

Im Falle des Directories wird hier noch ein wenig zeitsparender verfahren. Dazu wird 'f_frw' eine Null als Adresse des für die Übertragung benötigten Speicherbereichs übergeben. Daraufhin wird die Anfangsadresse der zu übertragenden Daten im Sektorpuffer zurückgeliefert (an Stelle der Zahl der übertragenen Zeichen), nachdem der Sektor über 'f_sread' geladen wurde. Das Kopieren der Daten aus dem Puffer heraus entfällt also.

Dies ist nur möglich, wenn die Daten garantiert alle innerhalb des gleichen Sektors liegen, da nach dem Laden des ersten Sektors auf jeden Fall abgebrochen wird. Bei Directory-Sektoren ist dies der Fall, da in jeden Sektor genau 16 Einträge (32*16=512 Byte) passen. Außerdem muß gewährleistet sein, daß die Daten auch möglichst bald verarbeitet werden, da unter Umständen schon beim nächsten Dateizugriff der Sektor aus der Pufferliste entfernt wird.

Bei der FAT ist dies nicht möglich, da durch das komplizierte Format ein Eintrag bei einer 12-Bit-FAT auch auf zwei Sektoren verteilt sein kann. Dies ist wohl ein Grund dafür, warum gerade FAT-Zugriffe bei GEMDOS sehr langsam sind.

Übrigens läßt sich dieser Spezial-Modus auch von eigenen Programmen aus verwenden, da man über 'Fread' direkten Zugang zu 'f_frw' hat. Da dies aber nicht dokumentiert ist und bei zukünftigen TOS-Versionen nicht mehr zu funktionieren braucht, sollt man davon absehen; außerdem gibt es wohl auch nur wenige Anwendungen, die davon Gebrauch machen könnten.

Größe von Sektoren und Clustern

Beim ST haben Sektoren immer eine Größe von 512 Byte, und Cluster bestehen immer aus zwei oder einem Sektor. GEMDOS erlaubt theoretisch auch andere Größen (jeweils Potenzen von 2), entsprechende Versuche führen allerdings nur zu Mißerfolgen.

Die entsprechenden Daten des BPB werden zwar korrekt interpretiert und intern wird mit ihnen auch richtig gerechnet, um Datei-Positionen usw. zu bestimmen.

Die Begrenzung der Sektorgröße liegt in der einheitlichen Größe der Sektorpuffer. GEMDOS kennt deren Größe nämlich nicht. Da jeder Puffer für Sektoren eines jeden Laufwerk in Frage kommt, müssen alle Puffer die im System maximal vorkommende Sektorgröße haben. Die vom BIOS installierten Puffer sind - wie nicht anders zu erwarten war - nur 512 Byte groß.

GEMDOS hat übrigens keine absoluten Zeiger auf BCBs oder Sektorpuffer. Daher ist eine Umgestaltung der Pufferlisten jederzeit möglich (natürlich nicht während der Abarbeitung einer GEMDOS-Funktion). Die Verwendung eines Massenspeichers, der mit 1024-Byte-Sektoren arbeitet, ist also durchaus möglich; das Treiberprogramm muß bei seiner Installation nur die Standard-Sektorpuffer gegen eigene der richtigen Größe austauschen. Man kann nur hoffen, daß es keine weiteren Schwierigkeiten gibt.

Cluster mit mehr als zwei Sektoren scheitern an einem Fehler in 'f_frw'. Ein Patch zur Behebung dieses Fehlers brachte allerdings nicht den erhofften Erfolg, so daß hierzu noch nicht das letzte Wort gesprochen ist.

Verwaltung der FAT

Zu den Routinen zur FAT-Verwaltung gibt es nicht allzuviel anzumerken. Sie sind in der Lage, Folge-Cluster, freie Cluster usw. zu ermitteln. Zum eigentlichen Zugriff auf die FAT wird 'f_frw' aufgerufen. Einzig und allein die Fehler machen wieder einmal zu schaffen. Der erste Fehler betrifft das Indizieren eines Eintrages in der FAT an Hand seiner Cluster-Nummer. Bei Cluster-Nummern größer als $3FFF (also nur bei 16-Bit-FATs) geht das schief. Bei zwei Sektoren pro Cluster ergibt sich somit die maximal erlaubte Größe eines Mediums zu 16 MB. Dieser Fehler wurde im Blitter-TOS sogar korrigiert (Cluster-Nummern bis $7FFF möglich)!! Allerdings möchte ich nicht dafür garantieren, daß Harddisk-Partitions bis 32 MB jetzt möglich sind, da es vielleicht weitere "Hindernisse" gibt.

Der zweite Fehler tritt nur bei 12-Bit-FATs auf. Er führt dazu, daß FAT-Einträge größer als $7FF falsch ausgewertet werden (GEMDOS arbeitet mit $Fxxx statt $0xxx weiter). Dies passiert aber nur bei ungeraden Clustern(!). Auch die Datei-Ende-Kennung $FFF wird nicht erkannt,

aber sie wird zufällig(!) richtig weiterverarbeitet. Bei zwei Sektoren pro Cluster wird somit "nur" die Kapazität auf 2 MB statt möglicher 4 MB beschränkt.
Beide Fehler resultieren übrigens aus für C typischen Vorzeichen-Fehlern.
Es sind nur $FFF bzw. $FFFF als Dateiende-Marke gedacht. Es gibt keine weiteren besonderen FAT-Einträge (wie $FF0-$FFE bei PC-DOS).

Nutzbarkeit der Sektor-Pufferung

Nach diesen Ausführungen über GEMDOS fragen Sie sich vielleicht schon verzweifelt, warum Sie (fast) noch nie etwas von der Pufferung bemerkt haben, warum z.B. beim Öffnen oder Schließen eines Ordners das Directory jedesmal neu von Diskette gelesen wird.
Die Ursachen hierfür sind über die verschiedensten Teile des Betriebssystems verstreut, fangen wir also beim BIOS an.
Beim Systemstart ist das BIOS für das Anlegen der (leeren) Pufferlisten verantwortlich. GEMDOS verwaltet diese nur, ändert aber nie die Größe der Listen. Da bei anderen Gelegenheiten kräftig mit dem Speicherplatz geaast wurde, sollte wohl hier ein wenig gespart werden. Beide Listen bestehen nämlich nur aus jeweils zwei Sektoren, wodurch der Nutzeffekt der Sektor-Pufferung fast verschwindet.
Dieser Umstand läßt sich relativ leicht beheben, da die Systemvariable 'bufl' legal zugänglich ist. Das Programm 'EXTBUFL' (Listing 2) erlaubt es, beide Listen beliebig zu erweitern. Es kann sogar mehrmals hintereinander gestartet werden, da es nur den für die Erweiterung benötigten Speicherplatz verbraucht und sich selbst vollständig freigibt.
Doch auch nach Einrichtung geradezu riesiger Pufferlisten wird Ihre Begeisterung sich in Grenzen halten.
Bei Disketten mit Schreibschutz ändert sich überhaupt nichts, da BIOS hier immer (sobald eine bestimmte Zeitspanne nach dem letzten Diskettenzugriff vergangen ist) einen "unsicheren Diskettenwechsel" meldet, so daß die Puffer wie bei 'f_sread' beschrieben immer neu geladen werden.
Bei Disketten ohne Schreibschutz werden Sie eine Verbesserung bemerken, wenn Sie sich z.B. in einer Fileselector-Box befinden. Hier können Sie nun fleißig Ordner auf- und zumachen, ohne sich über ein anlaufendes Floppylaufwerk zu ärgern.
Auf dem Desktop hat sich wiederum nichts geändert. Vor den meisten (vielleicht sogar allen) Diskettenoperationen "suggeriert" das Desktop dem BIOS nämlich einen "sicheren Diskettenwechsel"! Daraufhin gibt GEMDOS die gesamte Pufferliste frei, also Pufferung ade! Mit dieser zweifelhaften Methode wird über die Unzulänglichkeiten der BIOS-Diskettenwechsel-Erkennung hinweggetäuscht, wobei die Probleme jetzt auf die Anwenderprogramme verlagert werden, die nicht auf diese Scheinlösung zurückgreifen (wollen oder können). Dies erklärt, warum das Desktop so "hervorragend" (verglichen mit anderen Programmen) mit Diskettenwechseln zurechtkommt.
Nun könnten die Harddisk-Benutzer frohlocken, da es bei ihnen keine Medienwechsel gibt. Und siehe da, das Desktop macht keine Schwierigkeiten mehr.
Doch halt! Wer sich noch an die Beschreibung von 'f_swrite' erinnert, wird wissen, daß nicht veränderte Sektoren (was wohl der Normalfall sein dürfte) für ungültig erklärt werden. Nach jedem 'Fclose', welches alle Sektoren mit 'f_swrite' zurückschreibt, ist die Pufferliste somit leer. Das passiert unter anderem beim Kopieren und Laden von Programmen, so daß die Sektor-Pufferung auch bei Harddisk nur begrenzte Erfolge bringt.
Trotzdem kann es unter Umständen sinnvoll sein, die Pufferliste für die Benutzung bestimmter Programme zu vergrößern. Dabei sollten Sie vor allem die damit zusammenhängenden GEMDOS-Fehler beachten. Einige Programme mit raffinierten Dateizugriffen sorgen da womöglich für einige Überraschungen.
Falls Sie selbst ein wenig mit diesen Problemen herumspielen wollen, können Sie dies mit 'SHOWBUFL' (Listing 3) tun. Dieses Accessory zeigt den aktuellen Zustand der Pufferlisten an. Die Ausgaben stammen direkt aus den BCBs, bis auf 'sec'. Hierbei handelt es sich um die mittels des DMD aus der GEMDOS-Sektornummer errechnete BIOS-Sektornummer.
Nicht-Harddisk-Besitzer können auch mit einer RAM-Disk vorlieb nehmen, da für diese das Gleiche wie für Harddisk gilt, nur daß man von der Pufferung auf Grund ihrer Schnelligkeit nichts merkt.

Änderungen des TOS

In Abb. 5 sind Patches für TOS-Fehler angegeben. Dabei handelt es sich um die oben beschriebenen Fehler in der internen Funktion 'f_mrw' und der FAT-Verwaltung.
'f_mrw' macht nun alle mit 'f_swrite' geschriebenen Puffer ungültig, egal ob sie geändert waren oder nicht.

Schlußbemerkung

So, damit wären wir für heute mal wieder am Ende angekommen. Nächsten Monat beschäftigen wir uns u.a. mit der Verwaltung der Laufwerke, wozu auch die versprochene Erklärung des DMD gehört.

Alex Esser

```
          Adresse
RAM-TOS   ROM-TOS   ROM-TOS   Bytes (in Hex)
6.2.86    6.2.86    22.4.87

Fehler in 'f_mrw'
00b62a    fc579c    fc5a54    2a 78 04 b6 60 30 20 6e 00 12 30 28 00
                              06 b0 6d 00 04 66 20 30 2d 00 08 32 2e
                              00 0c b0 41 6d 14 d2 6e 00 0a b2 40 6f
                              0c 2e 8d 61 00 fe e4 3b 7c ff ff 00 04
00b663    fc57d5    fc5a8d    cc

Fehler bei 12-Bit-FAT
00bd03    fc5e75    fc612b    4f

Fehler bei 16-Bit-FAT
00bb98    fc5d0a    -------   7c 00 3c 07 e3 86 4e 71
```

Abb. 5: Patch für TOS-Fehler

```
/* Demonstrationsprogramm für Fehler in
   der Verwaltung der GEMDOS-Pufferliste */

#include <osbind.h>

#define NAME    "test"

main()
{  int fh;           /* Datei-Handle */
   int i;
   char c;
   char buf[512];

   if ((fh = Fcreate(NAME,0)) < 0)
      return;
         /* Datei lä_t sich nicht öffnen: Abbruch */
   c = 'a';
   for(i=0; i<512; i++)
        /* GEMDOS schreibt dies über Pufferliste */
      Fwrite(fh,1L,&c);
        /* (ein Zeichen würde auch reichen) */
   Fseek(0L,fh,0);
        /* zurück zum Dateianfang */
   for(i=0; i<512; i++)
        /* 'buf' voller 'b's schreiben */
      buf[i] = 'b';
   Fwrite(fh,512L,buf);
        /* GEMDOS schreibt dies direkt
           ...und lä_t Puffer so wie er war! */
   Fseek(0L,fh,0);      /* zurück zum Dateianfang */
   c = 'c';
        /* ein 'c' schreiben (über Pufferliste) */
   Fwrite(fh,1L,&c);
        /* Datei sollte jetzt 1 'c' und
           255 'b's enthalten, aber... */
   Fclose(fh);
     /*GEMDOS schreibt jetzt Puffer mit 'b's zurück*/
}
```

```
/* Erweiterung der GEMDOS-Pufferlisten
   24.1.1988 by A. Esser

   entwickelt mit MEGAMAX C */


#include <osbind.h>

#define bufl  (BCB **)0x4b2L
      /* Systemvariable 'bufl' */


/* Definition BCB */
typedef struct bctrl
{  struct bctrl *b_link;
   int b_bufdrv;
   int b_buftyp;
   int b_bufrec;
   int b_dirty;
   long b_dmd;
       /* korrekt: 'DMD *b_dmd', aber DMD
          hier nicht def. */
   char *b_bufr;
} BCB;


int install(buflp,nsec)
BCB **buflp;
int nsec;
{  long mp;
   int i;
   BCB *bcb;

   /* Speicherplatz für 'nsec' BCBs und
      Puffer reservieren */
   if ((mp = Malloc((long)nsec*
        (512+sizeof(BCB)))) == 0L)
   {  puts("Nicht genug Speicher.");
      return -1;
   }
   for (i=0; i<nsec; i++)
   {  bcb = (BCB *)mp;
           /* BCB initialisieren */
      bcb->b_bufdrv = -1;
           /* Puffer ungültig */
      bcb->b_buftyp = -1;
           /* kein gültiger Puffer-Typ */
      bcb->b_bufrec = 0;
           /* keine gültige Sektor-Nummer */
      bcb->b_dirty = 0;
           /* nichts geändert */
      bcb->b_dmd = 0L;
           /* BCB keinem Laufwerk zugeordnet */
      /* Puffer direkt hinter BCB legen */
      bcb->b_bufr = (char *)(mp+sizeof(BCB));
      /* BCB vorne in gewünschte
         Puffer-Liste einhängen */
      bcb->b_link = *buflp;
      *buflp = bcb;
      mp += 512+sizeof(BCB);
           /* weiter zum nächsten */
   }
   return 0;
}


main()
{  int nsec;
   long stack;
      /* gemerkter Supervisor-Stackpointer */

   stack = Super(0L);
      /* Supervisor-Mode notwendig */
   puts("\rWieviele zusätzliche FAT-Puffer?");
   scanf("%d",&nsec);
   install(bufl,nsec);
      /* Puffer für FAT nach Liste 'bufl[0]' */
   puts("\rWieviele zusätzliche DIR/DATA-Puffer?");
   scanf("%d",&nsec);
   install(bufl+1,nsec);
      /* Puffer für DATA/DIR nach Liste 'bufl[1]' */
   Super(stack);
      /* zurück nach User-Mode */
   /* Programm komplett freigeben, aber
      reservierten Speicher behalten */
   Ptermres(0L,0);
}
```

```
/* Accessory zur Anzeige der GEMDOS-Pufferlisten
   24.1.1988 by A. Esser

   entwickelt mit MEGAMAX C */


#include <osbind.h>
#include <gemdefs.h>
#include <obdefs.h>

#define bufl  (BCB **)0x4b2L /*Systemvariable 'bufl'*/


/* Definition DMD */
typedef struct
{  int d_roff[3];
   int d_drive;
   int d_fsiz;
   int d_clsiz;
   int d_clsizb;
   int d_recsiz;
   int d_numcl;
   int d_lclsiz;
   int d_mclsiz;
   int d_lrecsiz;
   int d_mrecsiz;
   int d_lclsizb;
   long d_fatfd;  /* korrekt: FD *d_fatfd */
   long d_dummy;
   long d_rdd;    /* korrekt: DD *d_rdd */
   int d_flag;
} DMD;

/* Definition BCB */
typedef struct bctrl
{  struct bctrl *b_link;
```

PADERCOMP — Walter Ladz

Erzbergerstraße 27 · 4790 Paderborn · Tel. 0 52 51-3 63 96

Floppystationen

PADERCOMP FL 1
3,5", 1 MB, NEC FD1037 A Laufwerk, anschlußfertig, graues Metallgehäuse, Abm: 153 x 103 x 26 mm **348.00**

PADERCOMP FL 2
Doppellaufwerk übereinander, sonst wie FL 1 .... **598.00**

PADERCOMP FL 3
Zweitlaufwerk für Amiga **348.00**

NEC FD 1037 A
3,5", 1 MB, 25,4 mm Bauhöhe, 5V **225.00**

Steckernetzteil
5V/1A, ideal für 2 NEC FD 1037 A **49.00**

Industrie-Floppystecker **7.90**

ST-Kabel an Shugart-Bus 3,5" **29.90**

Drucker

NEC P6 24 Nadeln, 216 Z/s, DIN A4 .... **1148.–**
NEC P7 24 Nadeln, 216 Z/s, DIN A3 .... **1498.–**
NEC P2200 24 Nadeln, P6-kompatibel **998.–**
Star LC-10 9 Nadeln, 144 Z/s, DIN A4 .... **598.–**
EPSON LQ-850 24 Nadeln, 264 Z/s, DIN A4 **1398.–**
EPSON LQ-500 24 Nadeln, 180 Z/s, DIN A4 **998.–**

Zubehör

FL-1-Adapter zum Anschluß von 2 FL-1 .... **49.00**
Multimatic Cut Sheet Feeder für P6 .... **598.00**
Dataphon S 21d-2 300 Baud, FTZ-Nr. **238.00**
Dataphon S 21/23 300 bzw. 1200/75 Baud, BTX ... **319.00**
Disk Box SS-50 für 50 Stück 3,5"-Disketten **19.90**
Media Box 1 für 150 Stück 3,5"-Disketten **34.90**
Druckerkabel ST **34.90**
Monitor-Ständer dreh-, schwenk- und kippbar **25.00**
Monitor-Stecker für ATARI ST **6.90**
Preisliste .... kostenlos

Monitore

EIZO Flexscan 8060 S alle Auflösungen, kein Nachjustieren, 820 x 620 Punkte, 0,28 dot pitch **1498.–**
NEC Multisync alle drei Auflösungen **1398.–**
Mitsubishi Freescan Modell 1471 .... **1448.–**
Kabel EIZO/NEC an ATARI ST **75.–**
Monitor-Umschaltbox **47.–**

Die aktuellen Preise erfragen Sie unter der PCB-Mailbox Tel. 0 52 52-34 13 (300 Bd., 8N1)

Bestellungen per Nachnahme ab 30 DM, Auslandslieferungen nur gegen Vorkasse. Eingetragenes Warenzeichen: ATARI ST. Die Preise können günstiger liegen. Rufen Sie an!

Händleranfragen erwünscht!

```
    int b_bufdrv;
    int b_buftyp;
    int b_bufrec;
    int b_dirty;
    DMD *b_dmd;
    char *b_bufr;
} BCB;


/* globale Variable für GEM */
extern int gl_apid;     /* Applikations-ID */
GRECT desk;             /* Desktop-Ma_e */
int idum;               /* int-Dummy */


/* Daten einer Pufferliste anzeigen */
show_bufl(bcb)
register BCB *bcb;
{  int btype;
   char stype[10];
   char srec[10];

   while (bcb)
   {  btype = bcb->b_buftyp;
      /* GEMDOS-Sektornummer aus BIOS-Sektornummer
         berechnen falls möglich */
      if (bcb->b_dmd)
      /* nur wenn Laufwerk vorhanden */
        sprintf(srec,"%d",bcb->b_dmd->d_roff[btype] +
               bcb->b_bufrec);
      else
        strcpy(srec,"-");
      switch (btype)
      {  case 0:
            strcpy(stype,"FAT");
            break;
         case 1:
            strcpy(stype,"DIR");
            break;
         case 2:
            strcpy(stype,"DATA");
            break;
         default:
            sprintf(stype,"%d",btype);
            break;
      }
      printf
 ("%06lx  %04x  %4s  %4d  %4s  %04x  %06lx  %06lx  \n",
   bcb, bcb->b_bufdrv, stype, bcb->b_bufrec,srec,
         bcb->b_dirty, bcb->b_dmd, bcb->b_bufr);
      bcb = bcb->b_link;
   }
}


do_work()
{  long stack;

   graf_mouse(M_OFF);            /* Maus ausschalten */
   Cconws("\033H\033B\033B");
       /* Cursor auf Beginn Zeile 2 */
   printf
 ("BCB     drv   type   rec   sec  dirty  DMD bufr     \n");
   stack = Super(0L);
       /* Supervisor-Mode notwendig */
   show_bufl(*bufl);
       /* Pufferliste für FAT ausgeben */
   printf("
                        \n");  /* 80 Spaces */
   show_bufl(*(bufl+1));
       /* Pufferliste für DIR/DATA ausgeben */
   Super(stack);
       /* zurück nach User-Mode */
   graf_mouse(M_ON);             /* Maus wieder an */
   Cnecin();
       /* Anzeige bis Tastendruck halten */
       /* Redraw des Desktop-Bildschirm über GEM */
   form_dial(FMD_FINISH,0,0,0,0,
          desk.g_x,desk.g_y,desk.g_w,desk.g_h);
}


main()
{  int msg[8];    /* Message buffer */
   int event;
   int menu_id;

   appl_init();
   if (gl_apid >= 0)
       /* 'appl_init' geglückt ? */
   {  wind_get(0,WF_WORKXYWH,
          &desk.g_x,&desk.g_y,&desk.g_w,&desk.g_h);
      menu_id = menu_register(gl_apid,
          "  Show buffer list");
      /* dafür sorgen, da_ printf-Puffer-Malloc unter
         AES geschieht */
      printf("\n");
   }
   while(1)                    /* Endlos-Schleife */
   {  event = evnt_multi(MU_MESAG,
          0,0,0,
          0,0,0,0,0,
          0,0,0,0,0,
        msg,0,0,&idum,&idum,&idum,&idum,&idum,&idum);
      if (event & MU_MESAG)
         if (msg[0] == AC_OPEN)
            /* nur AC_OPEN berücksichtigen */
            if (msg[4] == menu_id)
            /* eigenes Accessory? */
               do_work();           /* los geht's */
   }                               /* end of while */
}
```

ENDE



ABO



ABO

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte
mit
60 Pf.
frankieren

Heim-Verlag
Heidelberger Landstr. 194

6100 Darmstadt-Eberstadt
Telefon 0 61 51 / 5 60 57



Einzelheft- u. Monatsdisketten Bestellung



Einzelheft- u.
Disketten Service

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte
mit
60 Pf.
frankieren

Heim-Verlag
Heidelberger Landstr. 194

6100 Darmstadt-Eberstadt
Telefon 0 61 51 / 5 60 57



Kleinanzeigen



Kleinanzeigen

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte
mit
60 Pf.
frankieren

Heim-Verlag
Heidelberger Landstr. 194

6100 Darmstadt-Eberstadt
Telefon 0 61 51 / 5 60 57



Abonnement

Ja, bitte senden Sie mir die ATARI-Computer Fachzeitschrift ab ______________________
für mindestens 1 Jahr (11 Hefte) zum ermäßigten Preis von jährlich DM 70,– frei Haus.
(Ausland: Nur gegen Scheck-Voreinsendung DM 90,– Normalpost. DM 120,– Luftpost)
Der Bezugszeitraum verlängert sich nur dann um ein Jahr, wenn nicht 6 Wochen vor Ablauf des Abonnements gekündigt wird.

Name ______________________

Vorname ______________________

Straße/Nr. ______________________

PLZ ______ Ort ______________________

Gewünschte Zahlungsweise bitte ankreuzen

☐ Bequem und bargeldlos durch Bankeinzug

Konto-Nr. ______________ BLZ ______________

Institut ______________ Ort ______________

☐ Ein Verrechnungsscheck über DM______ liegt bei.

☐ Vorauskasse per Post-Einzahlung (Zahlkarte)

Garantie:
Diese Bestellung kann ich schriftlich innerhalb einer Woche (rechtzeitige Absendung genügt) widerrufen. Dies bestätige ich durch meine 2. Unterschrift.

Datum ______________ Unterschrift ______________

Datum ______________ Unterschrift ______________



ABO



Einzelheft- u. Monatsdisketten Bestellung

ST-Computer können Sie direkt beim HEIM-VERLAG zum Einzelheft-Preis von DM 6,— (1986 + 1987) DM 7,— (1988) nachbestellen. Bearbeitung nur gegen beigefügten Scheck über den entsprechenden Betrag (keine Überweisung).

Jan.	Febr.	März	April	Mai	Juni	Juli/Aug.	Sept.	Okt.	Nov.	Dez.	
											1986 = DM
											1987 = DM
											1988 = DM

\+ Gebühr für Porto u. Verp. = DM

(1 Heft DM 2,–, ab 2 Hefte DM 5,–)

☐ Scheck in Höhe zus. DM liegt bei

Disketten Service

Alle Programme, die in ST-Computer veröffentlicht wurden, sind auf Disketten erhältlich. Die Disketten enthalten die Programme von jeweils 2 ST-Computer-Ausgaben. Bestellen Sie durch ankreuzen die gewünschten Disketten

Preis je Diskette 28,– DM	Jan./Febr.	März/Apr.	Mai/Juni	Juli/Aug.	Sept./Okt.	Nov./Dez.	Jan./Febr.	März/Apr.	Mai/Juni
	87	87	87	87	87	87	88	88	88

Lieferung: gegen beigefügten Scheck zuzügl. 5,— DM Versandkosten, unabhängig der bestellten Menge.

COMPUTER

Einzelheft- u. Monatsdisketten Bestellung



Kleinanzeigen-Auftrag

Bitte veröffentlichen Sie für mich folgende Kleinanzeige in der angekreuzten Rubrik

Biete an ☐ Hardware ☐ Software **Ich suche** ☐ Hardware ☐ Software ☐ **Tausch** ☐ **Kontakte** ☐ Verschiedenes ☐

30 Buchstaben je Standardzeile – incl. Satzzeichen und Wortzwischenräume.
Groß- und Kleinbuchstaben verwenden, fettgedruckte Wörter unterstreichen.

Bearbeitung **nur gegen Vorausscheck** über den entsprechenden Betrag (keine Überweisung)

☐ privat = DM 7,– je Zeile incl. MwSt.
☐ gewerblich = DM 15,– je Zeile + MwSt.
☐ Chiffregebühr = DM 10,–

☐ Scheck über DM______ ist beigefügt

Bei Angeboten: Ich bestätige, daß ich alle Rechte an den angebotenen Sachen besitze.

Datum ______________ Unterschrift ______________

Absenderangaben auf der Rückseite nicht vergessen

COMPUTER

Kleinanzeigen



Kontaktkarte

Kontaktkarte

Bitte Adresse der Firma, bei der Sie Informationen, oder etwas bestellen möchten, auf der rechten Seite eintragen → → → → → → → → → →

Absender
(Bitte deutlich schreiben)

Vorname/Name

Beruf

Straße/Nr.

PLZ/Ort

Telefon Vorwahl/Rufnummer

Postkarte

Bitte freimachen

Firma

Straße/Postfach

PLZ Ort



Kurzmitteilung



Kurzmitteilung

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Telefon

Postkarte

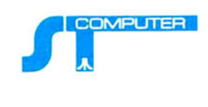
Bitte freimachen

Merlin Computer GmbH
ST-Computer Redaktion
Industriestraße 26

6236 Eschborn



PD Bestellung



PD Bestellung

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte freimachen

Merlin Computer GmbH
ST-Computer Redaktion
Industriestraße 26

6236 Eschborn

COMPUTER

☐ Ich bitte um weitere Informationen

☐ Ich gebe folgende Bestellung auf

in Bezug auf Ihre Anzeige in ST-Computer Heft ______ Seite ___

Menge	Produkt/Bestellnummer	DM	gesamt DM

Datum, Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

Abgesandt am:

COMPUTER

Firma:

Bemerkungen:

COMPUTER

Meine Meinung

Zu dem Artikel ____________ in Heft ____, Seite ____ hätte ich folgendes zu bemerken:

☐ Ich möchte Ihnen folgendes Programm zur Veröffentlichung anbieten: (Kurzbeschreibung, Sprache, Länge in Druckerseiten, GEM/TOS)

☐ Ich kann über folgendes Thema berichten: (Tips & Tricks am ST, Hardware, Software, etc.)

☐ Ich möchte gerne Autor in der ST-Computer werden. Meine Fachgebiete: (z.B. LISP, Pearl, Modula-2, Assembler ...)

☐ Ich möchte, daß folgendes Public-Domain Programm in Ihre Sammlung aufgenommen wird.

☐ Sonstiges

Bei weiteren Angaben oder Fragen wenden Sie sich bitte schriftlich oder telefonisch an die Redaktion. Tel. 0 61 96/48 18 11

COMPUTER

Kurzmitteilung

COMPUTER

PUBLIC DOMAIN SERVICE

Ich bestelle folgende PD-Disketten:
(Siehe PD Service in dieser Ausgabe)

Zahlung erfolgt:

☐ per Scheck

☐ per Nachnahme

Je Diskette fügen Sie bitte einen Betrag von DM 10,– bei,
für Porto und Verpackung je Sendung DM 5,– (Ausland DM 10,–)

Datum

Unterschrift

COMPUTER

PD Bestellung

Datum

Einkaufsführer

Hier finden Sie Ihren Atari Fachhändler

Anzeigenschluß für Heft 5/88: 25.03.88

2210 Itzehoe

Der Computerladen
Inhaber Ulrich Bübel Martin Kopplow

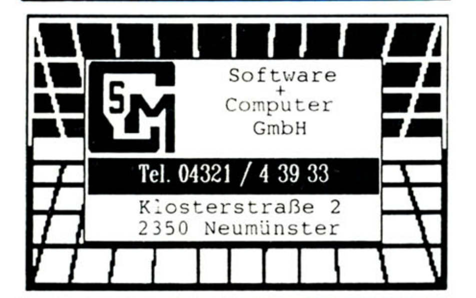
Coriansberg 2 · 2210 Itzehoe
Telefon (0 48 21) 33 90/91

2300 Kiel



Die Welt der Computer
Dreiecksplatz Nr. 7
2300 Kiel 1 · ☎ 04 31/56 70 42

2350 Neumünster



2390 Flensburg

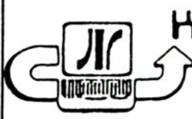


2800 Bremen

PS-DATA
Doventorsteinweg 41
2800 Bremen
Telefon 04 21 - 17 05 77

2850 Bremerhaven

HEIM- UND PERSONALCOMPUTER



Hurt Neumann
„Bürger" 160
2850 Bremerhaven
Tel. 04 71/4 20 06

HARDWARE · SOFTWARE · PAPIERWARE

2940 Wilhelmshaven

Radio Tiemann
ATARI-Systemfachhändler

Markstr. 52
2940 Wilhelmshaven
Telefon 0 44 21 - 2 61 45

2950 Leer



- HARDWARE-SOFTWARE
- SYSTEM-ENTWICKLUNG
- ORGANISATION
- EDV-SCHULUNG
- EDV-BERATUNG
- SERVICE-WARTUNG

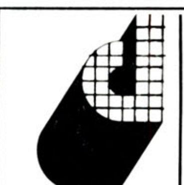
Augustenstraße 3 · 2950 Leer
Telefon 04 91 - 45 89

3000 Hannover

COM DATA

Am Schiffgraben 19 · 3000 Hannover 1
Telefon 05 11 - 32 67 36





DATALOGIC COMPUTERSYSTEME
ATARI ST · BERATUNG
COMPUTER · SERVICE
HARDWARE · VERKAUF
SOFTWARE
CALENBERGER STR 26
3000 HANNOVER 1
TEL 0511 · 32 64 89

trendDATA Computer
IBM · EPSON · TRIUMPH ADLER
HEWLETT PACKARD · ATARI etc
trendDATA Computer GmbH
Am Marstall 18–22 · 3000 Hannover 1
Telefon (05 11) 1 66 05-0

3040 Soltau

F & T Computervertrieb

Am Hornberg 1
(Industriegeb. Almhöhe)
3040 Soltau
Tel. 0 51 91/1 65 22

3150 Peine

Wieckenberg & Schrage GmbH
Computertechnik
Hard- u. Software

Woltorfer Str. 8, 3150 Peine
Tel. 0 51 71/60 52/3 o. 0 51 73/79 09

3170 Gifhorn

COMPUTER-HAUS
GIFHORN
Braunschweigerstr. 50
3170 Gifhorn
Telefon 0 53 71 - 5 44 98

3300 Braunschweig



Rebenring 49-50
3300 Braunschweig
Tel. (05 31) 33 32 77/78

3400 Göttingen

Büroeinrichtungs-Zentrum
Wiederholdt

3400 Göttingen-Weende
Wagenstieg 14 – Tel. 05 51/38 57-0

3470 Höxter



3500 Kassel

Hermann Fischer GmbH
autorisierter ATARI-Fachhändler

Rudolf-Schwander-Str. 5-13
3500 Kassel
Tel. (05 61) 70 00 00

3550 Marburg

L W M
COMPUTER SERVICE
Bahnhofstraße 26b
3550 Marburg/Lahn
✆ 0 64 21 - 6 22 36

4000 Düsseldorf

BERNSHAUS GmbH
Bürotechnik – Bürobedarf
Cäcilienstraße 2
4000 Düsseldorf 13 (Benrath)
Telefon 02 11 - 71 91 81

4000 Düsseldorf

HOCO
EDV ANLAGEN GMBH
Flügelstr. 47
4000 Düsseldorf
Tel. 02 11 - 77 62 70

4050 Mönchengladbach



Hindenburgstr. 249
4050 Mönchengladbach
Tel. 0 21 61 - 18 7 64

4150 Krefeld



- Service-Center
- Hardware
- Software
- Erweiterungen

COP
Computer Service GmbH
Lewerentz Str. 111
4150 Krefeld
Tel. 0 21 51 / 77 30 42

4250 Bottrop

Megateam-Computer-Systeme

Kirchhellenerstraße 262
4250 Bottrop

4300 Essen

ATARI Systemfachhändler



KARSTADT Aktiengesellschaft
Limbecker Platz 4300 Essen 1
Tel.: (02 01) 17 63 99

4320 Hattingen



Hattingen, Große Weilstr. 18-20, Telefon (0 23 24) 20 94 77

4330 Mülheim



Computer und Bürotechnik
Vertriebsgesellschaft mbH
Dickswall 79 4330 Mülheim Telefon 02 08 / 3 40 34

Computer Hard- und Software auch im Leasing
Computerkurse für Anfänger und Fortgeschrittene

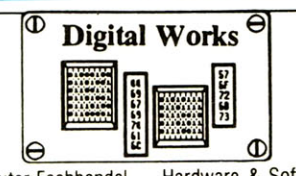
Tandon Computer · VICTOR · SEL-Fernkopierer · NEC · ATARI · OKI

4400 Münster



COMPUTER SYSTEME GMBH
Daimlerweg 39 - 4400 Münster
Telefon 02 51 / 71 99 75 - 9

4410 Warendorf



Computer-Fachhandel – Hardware & Software

Jörg Kellert – Helmut Müller GbR
Brünebrede 17 · 4410 Warendorf
Tel. 0 25 81 / 6 11 26

4422 Ahaus

ATARI · Epson · Fujitsu Molecular · NCR · Tandon · Schneider · Star

OCB

OCB-Computershop Wallstraße 3 4422 Ahaus Tel. 0 25 61/50 21	OCB-Hard- und Software Wessumerstraße 49 4422 Ahaus Tel. 0 25 61/50 21

4430 Steinfurt

ATARI SCHNEIDER STAR NEC SEIKOSHA PANASONIC EPSON

Computer
Büromaschinen
Service

Telefon 02551/2555

Tecklenburger Str. 27 · 4430 Steinfurt

4500 Osnabrück

Heinicke-Electronic
Kommenderiestr. 120·4500 Osnabrück
Telefon 05 41 - 8 27 99
Wir liefern Micro-Computer seit 1978

4600 Dortmund

Bürostudio BOLZ
Brauhausstraße 4 · 4600 Dortmund
Telefon 02 31 - 52 77 13 - 16



4600 Dortmund



Atari, Genie, Schneider, Tandy, Brother, Star, Memorex, BASF, Verbatim
cc Computer Studio GmbH
Software-Hardware-Beratung
Service-Eilversand

Ihre Ansprechpartner:
v. Schablinski
Jan P. Schneider T. 0231/528184 · Elisabethstraße 5, 4600 Dortmund 1 · Tx 822631 cccsd

Elektronik
Computer
Fachliteratur

ATARI-System-Fachhändler

4600 Dortmund 1, Güntherstraße 75, Tel. (02 31) 57 22 84



4620 Castrop-Rauxel

R. Schuster Electronic
OBERE MÜNSTERSTR. 33-35 · (02305) 37700 · 4620 CASTROP-RAUXEL



System-Fachhändler

4650 Gelsenkirchen-Horst



Hard- und Software, Literatur
Bauteile, Service, Versand
Groß- und Einzelhandel
Poststr. 15 · 4650 Gelsenkirchen-Horst
Tel. 0209/52572

4700 Hamm

computer center



4800 Bielefeld

hardware
software
organisation
service

CSF

CSF COMPUTER & SOFTWARE GMBH
Heeper Straße 106 – 108
4800 Bielefeld 1
Tel. (05 21) 6 16 63

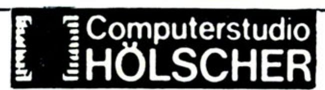
Carl-Severing-Str. 190
4800 Bielefeld 14
Telefon: 05 21 / 45 99-150
Telex : 9 37 340 krab d
Telefax: 05 21 / 45 99-123

MICROTEC
Software
Hardware
Beratung
Service

5000 Köln

BÜROMASCHINEN
braun
AM RUDOLFPLATZ GmbH
5000 KÖLN 1
RICHARD-WAGNER-STR. 39
RUF: 0221/219171

5010 Bergheim



EDV-Beratung · Organisation
Programmierung · Home/Personal-Computer
Software · Zubehör · Fachliteratur
Zeppelinstr. 7 · 5010 Bergheim
Telefon 02271-62096

5060 Bergisch-Gladbach

Computer Center
Buchholzstraße 1
5060 Bergisch-Gladbach
Telefon 02202-35053

5090 Leverkusen

Rolf Rocke
Computer-Fachgeschäft
Auestraße 1
5090 Leverkusen 3
Telefon 02171/2624

5200 Siegburg

Computer Center
Luisenstraße 26
5200 Siegburg
Telefon 02241/66854

5220 Waldbröl



Waldstraße 1 · 5220 Waldbröl
☎ (02291) 4408/3386

5300 Bonn



Gesellschaft für Computer- und Communicationstechnologie mbH
Hardware · Software · EDV-Zubehör
Telefon 0228/222408
COCO GmbH · Schumannstraße 2 · 5300 Bonn 1

5457 Straßenhaus

DR. AUMANN GMBH
Computer-Systeme
Schulstr. 12
5457 Straßenhaus
Telefon 02634-4081/2

5500 Trier



Güterstr. 82 - 5500 Trier
☎ 0651-25044
Fordern Sie unsere Zubehör-Liste an.

5540 Prüm

ATC COMPUTER
J. M. ZABELL
Ritzstraße 13 · Pf. 1051
5540 PRÜM
– Tel.: 06551-3039 –

5600 Wuppertal

Jung am Wall
Wall 31—33
5600 Wuppertal 1
Telefon 0202/450330

MEGABYTE
Computer Vertriebs GmbH
Friedrich-Engels-Allee 162
5600 Wuppertal 2 (Barmen)
Telefon (0202) 81917

5630 Remscheid

COM SOFT
Scheiderstr. 12 · 5630 Remscheid
Telefon (02191) 21033-34

5800 Hagen



Vertragshändler **Axel Böckem**
Computer + Textsysteme
Eilper Str. 60 (Eilpezentrum) · 5800 Hagen
Tel. 02331/73490

5900 Siegen



Siegen · Weidenauer Str. 72 · ☎ 0271/73495

6000 Frankfurt

Müller & Nemecek
Kaiserstraße 44
6000 Frankfurt/M.
Tel. 069-232544

WAIZENEGGER
Büroeinrichtungen
Kaiserstraße 41
6000 Frankfurt/M.
☎ 069/27306-0

ATARI
... wir machen Spitzentechnologie preiswert.
Jetzt bei uns. **Vertragshändler**



Büro-Computer + Organisations GmbH
Oederweg 7–9
6000 Frankfurt/M. 1
☎ (069) 55 04 56-57
Hardware ★ Software ★ Beratung ★ Service

6100 Darmstadt

Heim
Büro- und Computermarkt
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
✆ 06151/56057

ATARI Systemfachhändler



KARSTADT Aktiengesellschaft
Elisabethenstr. 15 · 6100 Darmstadt
Luisencenter · Tel. 06151-109420

6240 Königstein

KFC
COMPUTERSYSTEME
Wiesenstraße 18
6240 Königstein
Tel. 06174-3033
Mail-Box 06174-5355

6300 Gießen



6300 Gießen

Schneider ATARI Commodore
BAUMS
BÜRO · ORGANISATION
Bahnhofstr. 26 · 6300 Gießen
Telefon: 06 41 / 7 10 96



Thomas Heß
Asterweg 10 · 6300 Gießen
Telefon 06 41 / 3 91 53

6330 Wetzlar



Fachmarkt für Computer u. Unterhaltungs-electronic in Wetzlar,
Einkaufszentrum Bahnhofstraße, Tel. (0 64 41) 4 85 66

6400 Fulda

Schneider ATARI Commodore
WEINRICH
BÜRO · ORGANISATION
Ronsbachstraße 32 · 6400 Fulda
Telefon: 06 61 / 4 92 - 0

6457 Maintal

Landolt-Computer
Beratung · Service · Verkauf · Leasing
Wingertstr. 112
6457 Maintal/Dörnigheim
Telefon 0 61 81 - 4 52 93

6500 Mainz

:ELPHOTEC
Computer Systeme
Ihr Atari Systemhändler
mit eigenem Service-Center
Walpodenstraße 10
6500 Mainz
Telefon 0 61 31 - 23 19 47

6520 Worms

ORION
Computersysteme GmbH
Friedrichstraße 22
6 5 2 0 W O R M S
Tel. 0 62 41 / 67 57 - 67 58

6700 Ludwigshafen

MKV Computermarkt
Bismarck-Zentrum
6700 Ludwigshafen
Telefon 06 21 - 52 55 96

6720 Speyer

MKV Computermarkt
Gilgenstraße 4
6720 Speyer
Telefon 0 62 32 - 7 72 16

6750 Kaiserslautern

C.O.S
Computer
Organisation GmbH
Friedrichstr. 4 · 6750 Kaiserslautern
Telefon (06 31) 9 20 51

6800 Mannheim

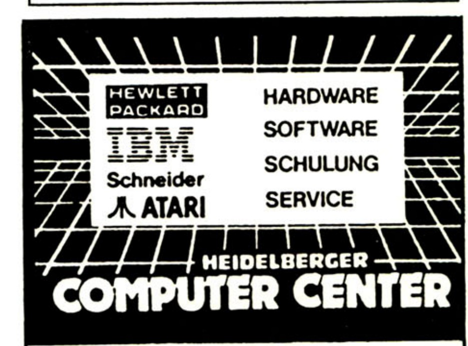


Computersysteme + Textsysteme
6800 Mannheim 24
Casterfeldstraße 74-76
☎ (06 21) 85 00 40 · Teletex 6 211 912

Computer-Center
am Hauptbahnhof GmbH
L 14, 16-17
6800 Mannheim 1
Tel. (06 21) 2 09 83 / 84

6900 Heidelberg

JACOM COMPUTERWELT
Hardware · Software
Schulung · Service
Mönchhofstraße 3 · 6900 Heidelberg
Telefon 0 62 21 / 41 05 14 - 550



Bahnhofstr. 1 / Tel. 0 62 21 / 2 71 32

7022 L-Echterdingen

Autorisierter ATARI-System-Fachhändler
ATARI ST



Matrai Computer GmbH
Bernhauser Str. 8
7022 L.-Echterdingen
☎ (07 11) 79 70 49

7030 Böblingen

Verkauf – Service – Software



COMPUTER CENTER
Norbert Hlawinka
Sindelfinger Allee 1
7030 Böblingen
Tel. 0 70 31 / 22 60 15

7100 Heilbronn

Unser Wissen ist Ihr Vorteil
Walliser & Co.
Mönchseestraße 99
7100 Heilbronn
Telefon 07131/60048



Computer-Welt
Am Wollhaus 6
7100 Heilbronn
Tel. 0 71 31 - 6 84 01 - 02

7101 Flein

der **COMPUTERLADEN** von



in der Falterstraße
7101 Flein
Beratung, Service, Verkauf,
Software-Entwicklung
direkt beim Peripherie-Hersteller

7150 Backnang



7400 Tübingen

Werner Brock
COMPUTERSSTUDIO
Poststraße 2 – 4 · D-7400 Tübingen
Tel. 0 70 71 - 3 43 48 · Fax 0 70 21 - 3 47 92
Autorisierter Systemfachhändler für:
ATARI, Schneider, Commodore, Panasonic, Kaypro, Sharp, NEC, OKI, STAR,...

7410 Reutlingen

Werner Brock
COMPUTERSHOP
Federnseestr. 17 · 7410 Reutlingen
Tel. 07121-342 87
Tx 172 414 024 RMI D · box:rmi:taisoft · Fax 07121-33 97 79
Autorisierter Systemfachhändler für:
ATARI, Schneider, Commodore, Panasonic, Kaypro, Sharp, NEC, OKI, STAR,...

7450 Hechingen



Gesellschaft für Datenverarbeitung mbh
Computer · Drucker
Zubehör · Fachliteratur
Schloßplatz 3 · 7450 Hechingen
Telefon 07471/14507

7475 Meßstetten

Ihr ATARI-Systemhändler im Zollern-Alb-Kreis
HEIM + PC-COMPUTERMARKT
HARDWARE · SOFTWARE · LITERATUR


ATARI COMMODORE CUMANA DATA-BECKER
MULTITECH RITEMAN SCHNEIDER THOMSON
7475 Meßstetten 1 · Hauptstraße 10 · 07431/61280

7480 Sigmaringen

Wir stellen uns für Sie auf den Kopf!
Wir führen:
Hardware - Software - Zubehör
Zeitschriften - Bücher
Wir sind autorisierte Fachhändler von:


SOFT&EASY
COMPUTER GMBH
7480 Sigmaringen, Am Rappgassle, Tel. 07571/12483
7968 Saulgau, Pfarrstraße 13, Tel. 07581/2598

7500 Karlsruhe

papierhaus erhardt
Am Ludwigsplatz · 7500 Karlsruhe
Tel. 0721-1608-16

MKV GMBH
Kriegsstraße 77
7500 Karlsruhe
Telefon 0721-84613

7530 Pforzheim

DM Computer GmbH
Hard- & Software
Durlacherstr. 39
Tel. 07231-13939
7530 Pforzheim
Telex 783248

7600 Offenburg

FRANK LEONHARDT ELECTRONIC
Ihr Fachgeschäft für Microcomputer · Hifi · Funk
In der Jeuch 3
7600 Offenburg
Telefon 0781/57974

7640 Kehl/Rhein



Computer-Software-Marketing
eigener Service · eigene Software
Badstrasse 12
Tel. 07807/822
Telex: 752913
7607 NEURIED 2
Filiale:
Hauptstrasse 44
Tel. 07851/1822
7640 KEHL/RHEIN
ELEKTRO-MÜNTZER GmbH

7700 Singen

U. MEIER
Computersysteme
7700 Singen-Htwl.
Am Posthalterswäldle 8
Telefon 07731-44211

7730 VS-Schwenningen

BUS BRAUCH & SAUTER COMPUTER TECHNIK
Villinger Straße 85
7730 VS-Schwenningen
Telefon 07720/38071-72

7750 Konstanz

ATARI ★ PC's ★ SCHNEIDER
computer-fachgeschäft
rösler
Rheingutstr. 1 · 07531-21832

7800 Freiburg

CDS
EDV-Service GmbH
Windausstraße 2
7800 Freiburg
Tel. 0761-81047

7800 Freiburg



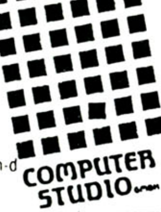
KARTÄUSERSTRASSE 59
D-7800 FREIBURG/BRST.
TELEFON 0761-382038

7890 Waldshut-Tiengen

hettler-data
service gmbh
Lenzburger Straße 4
7890 Waldshut-Tiengen
Telefon 07751/3094

7900 Ulm

HARD AND SOFT COMPUTER GMBH
Ulms großes Fachgeschäft
für BTX, Heim- u. Personalcomputer
Herrenkellergasse 16 · 7900 Ulm/Donau
Telefon 0731/62699

EDV-Systeme
Software-erstellung
Schulung
Systemhaus:
Frauenstr. 28
7900 Ulm/Donau
Tel. 0731/28076
Telex 712973 csulm-d


7918 Illertissen

biotech gmbh
technische Informationssysteme
Computerladen
Marktplatz 13
7918 Illertissen
07303/5045

7950 Biberach

HARD AND SOFT COMPUTER GMBH
Biberachs großes Fachgeschäft
für BTX, Heim- u. Personalcomputer
Schulstraße 6 · Bei der VHF
7950 Biberach · Tel. 07351/12221

7980 Ravensburg

GRAHLE
Expert Grahle Computer
Eisenbahnstr. 33
7980 Ravensburg
Tel.: 0751/15955
Vertragshändler für ATARI, Schneider und Star

8000 München



COMPUTER + BÜROTECHNIK

COMPUTER · SOFTWARE · PERIPHERIE
BERATUNG · TECHN. KUNDENDIENST
INGOLSTÄDTER STR. 62 L
EURO-INDUSTRIE-PARK · 8000 MÜNCHEN 45
TELEFON 089/3113066 · TELETEX 898341

schulz computer

Schillerstraße 22
8000 München 2
Telefon (0 89) 59 73 39

Beratung · Verkauf · Kundendienst

8032 Gräfeling



COMPUTER SYSTEME SCHULUNG
Am Haag 5
8032 Gräfelfing
Tel. 089-8545464,851043

8100 Garmisch-Partenk.

Uwe Langheinrich
Elektronik Center
Hindenburgerstr. 45
8100 Garmisch-Partenkirchen
Tel. 0 88 21 - 7 15 55
Bitte Gratisliste anfordern

8120 Weilheim

Klement

COMPUTER FACHHÄNDLER

Elektro-, Radio- und Fernseh-Center
Beleuchtungskörper · Schallplatten
Meisterbetrieb

8120 Weilheim · Admiral-Hipper-Straße 1
Geschäft ✆ 45 00 · Kundendienst ✆ 44 00
Computer-Studio ✆ 6 21 67
Interfunk-Fachgeschäft

8150 Holzkirchen

ATARI
Besuchen Sie uns!
Fordern Sie unseren Software-Katalog (520ST) an!



MÜNZENLOHER GMBH
Tölzer Straße 5
D-8150 Holzkirchen
Telefon: (0 80 24) 18 14

8170 Bad Tölz

Uwe Langheinrich
Elektronik Center
Wachterstr. 3
8170 Bad Tölz
Tel. 0 80 41 - 4 15 65
Bitte Gratisliste anfordern

8330 Eggenfelden

Hot Space

Computer-Centrum
R. Lanfermann

Schellenbruckstraße 6
8330 Eggenfelden
Telefon 0 87 21 / 65 73

Altöttinger Straße 2
8265 Neuötting
Telefon 0 86 71 / 7 16 10

Innstraße 4
8341 Simbach
Telefon 0 85 71 / 44 10

8400 Regensburg

Zimmermann
elektroland

8400 Regensburg
Dr.-Gessler-Str. 8
☎ 09 41 / 9 50 85

8390 Passau
Kohlbruck 2a
☎ 08 51 / 5 20 07

8423 Abensberg

COMPUTERVERSAND
WITTICH
Tulpenstr. 16 · 8423 Abensberg
☎ 0 94 43 / 453



8500 Nürnberg

EINE IDEE ANDERS
KARSTADT NÜRNBERG AN DER LORENZKIRCHE
TECHNIK CENTER
1. KLASSE EINKAUFEN IM WELTSTADTHAUS



8520 Erlangen

Ihr Partner
IN EDV-FRAGEN
Wir führen prof.
HARD-& SOFTWARE
für
IBM PC/XT/AT
APPLE II+/e
ATARI 520+1040
AMIGA



☎ 09131/22600

ALPHATRON



8520 Erlangen

Computerservice
Decker
Meisenweg 29 - 8520 Erlangen
Telefon 0 91 31 / 4 20 76

Zimmermann
elektroland

8520 Erlangen
Nürnberger Straße 88
Tel. (0 91 31) 3 45 68

8500 Nürnberg
Hauptmarkt 17
Tel. (09 11) 2 07 98

8600 Bamberg



Bamberg · Tel. 0951 / 27808 - 09

8700 Würzburg

SCHOLL
BÜROTEAM

Hardware · Software
Service · Schulung

computer center
am Dominikanerplatz
Ruf (09 31) 5 04 88

8720 Schweinfurt

Uhlenhuth GmbH
Computer + Unterhaltungselektronik
Albrecht-Dürer-Platz 2
8720 Schweinfurt
Telefon 0 97 21 / 65 21 54

8900 Augsburg



Unser Plus: Beratung u. Service
Schwalbenstr. 1 · 8900 Augsburg-Pfersee
Telefon (08 21) 52 85 33 oder 52 80 87
Computer Vertriebs- und Software GmbH

8900 Augsburg

ATARI Systemfachhändler



KARSTADT Aktiengesellschaft
Bürgermeister-Fischer-Str. 6-10
8900 Augsburg · Tel. (08 21) 31 53-416

8940 Memmingen

EDV-Organisation
Hard- + Software
Manfred Schweizer KG
Ulmer Str. 2, Tel. 08331/12220
8940 Memmingen

Österreich

A-1020 Wien

W + H.
Computerhandel Ges m.b.H.

Förstergasse 6/3/2 · 1020 Wien
Tel. 02 22-35 09 68

A-1040 Wien

Ihr ST-Fachhändler in Wien
Computer-Studio
Wehsner Gesellschaft m.b.H.

1040 Wien - Paniglgasse 18-20
Telefon 02 22-65 78 08, 65 88 93

A-8010 Graz



Schweiz

CH-1205 Geneve

PIMENT ROUGE
INFORMATIQUE S.A.
8, RUE DES MARAICHERS
1205 GENEVE TEL. 022/28 56 24

CH-1700 Fribourg

FRIDAT SA INFORMATIQUE
ehem.
Softy Hard's Computershop
Rte des Grives 4
1700 Granges-Paccot / Fribourg
0041 (0)37 26 66 28

CH-2503 Biel

UE
URWA Electronic

Ihr ATARI ST Spezialist in
der Schweiz.
0 32 / 25 45 53

Lindenweg 24, 2503 Biel

CH-3084 Bern

Computer Corner
Ihr ATARI ST-Partner in Bern

Hard-Software · Beratung
Midi-Schulung · Service
Seftigenstr. 240 · 3084 Bern-Wabern
Zentrum Wabern · Tel. 031/54 51 00

CH-4054 Basel

Wir sind Amiga- und ST-Freaks.

Unsere Öffnungszeiten:
Von Dienstag bis Freitag
9.30 – 12.30 und 14.00 – 18.30 und
am Samstag 9.30 – 16.00

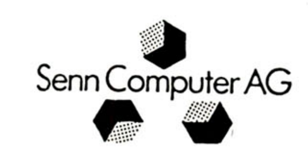
SYSAG
COMPUTER CENTER
Basel Tel. 061/39 25 25 · Holeestrasse 87 · 4054 Basel
Aarau Tel. 064/22 63 33 · Kasernenstrasse 26 · 5000 Aarau

CH-4625 Oberbuchsiten

STECTRONIC M. Steck
Electronic-Computer-Shop

Hauptstr. 104/137
CH-4625 OBERBUCHSITEN
Tel. 062/63 17 27 + 63 10 27

CH-5430 Wettingen



Zentralstrasse 93
CH-5430 Wettingen
Tel. 056 / 27 16 60
Telex 814 193 seco

CH-8001 Zürich

ADAG LASERLADEN
SEILERGRABEN 41. 8001 ZÜRICH
MO-FR: 12-18.30 TEL. 01/251 49 34
DTP... LASERPRINTS ... ET CETERA!

CH-8006 Zürich

ADAG
COMPUTER-SHOP
UNIVERSITÄTSSTR. 25
8006 ZÜRICH TEL. 01/252 18 68

Computer-Center P. Fisch

Stampfenbachplatz 4
8006 ZÜRICH
✆ 01/363 67 67

CH-8021 Zürich



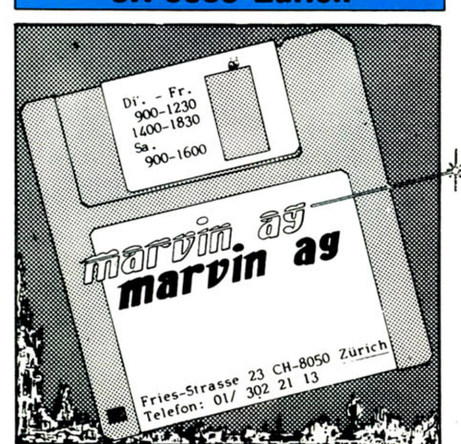
Langstrasse 31
Postfach
CH-8021 Zürich
Tel. 01 / 241 73 73
Telex 814 193 seco

CH-8021 Zürich

VILAN
Das Warenhaus der neuen Ideen

Ihr Computer-Fachhandel an der
Bahnhofstrasse 75 · Zürich

CH-8050 Zürich



Kleinanzeigen

James Seite 51

BIETE HARDWARE

Drucker Fujitsu DX 2100 mit reichlich Zubehör DM 2250,– Dataphon S 21d DM 150,– 3,5" Diskettenkasten Stück DM 35,–. Tel. 04151-81131 ab 16.30 h

★ ★ FESTPLATTE 44 MB! ★ ★ Modell ICF für alle ST/MEGA inkl. neues TOS-ROM, Garantie. PREISIDEE: 2.000,– DM (VHB!) Telefon: 0631/13191

Verk. HF-Modulator 150 DM (NPR 298 DM) 06131/572794

CRP-Digitalisiertablett A4 neuwertig 800,– DM 089/8545696

Epson FX 85 800 DM 05462/1808

MEGA ST 4, Garantie bis 1.3.88 Bestzust., wg. Systemw. 3200 DM Tel.: 06131/393563 Martin

Atari SC 1224 Farbmonitor 1 Mt., umstdh. abzugeben VB 700,– F. Waldschmidt 06151/539146 abd.

260 ST, 1 MB, SF354, SM124, Maus Disketten, Bücher 1200,– 09438/661 ab 17.00 Uhr

Verkaufe neuw. Video-Interface Box-Atari mit Netzgerät DM 150 Jürgen Hermann Frühmessgarten 14, 7277 Wildberg

HD. Atari 980,– Tel. 0911/649573

Hard-Disk 30 MB - Neu: 1098 DM **Uhr** - ST: 25 DM! **Eprom-Disk:** 60 DM! J. Lazaridis, Blumberg-48, 4200 Oberhausen 1: Gratisinfo! 853274

520/1040/SM124/usw. Superpreise Tel. 05732-7743 ab 17.00 Uhr

Epson FX105, Profidrucker, DINA3 Einzelblattautomat 02266/7894

Universelle Experimentierkarte (Europa-Format) mit Bus-Interface für **ATARI MEGA-ST** zum Einstecken in internen Busanschluß. Großes Verdrahtungsfeld von 36x47 Lötaugen durchkontaktiert, 8-Chip-Select Signale, A1–A5, Clock, Reset, R/W gepuffert. 8 Bit breiter gepufferter Datenbus. **Nur** 4TTL-Chips. Komplettbestückung mit Stecker. DM 128,– Noch Fragen??? **Tel.: 07121-60298/99.**

ST 520+, Rom-Tos, SF314, SM124, Softw. wg. Syst.wechsel zu verk. Tel. 02302/71718 od. 02324/83057

Sparen Sie 400 DM! Festplatten für Atari ST ab 900 DM (20 MB-Seagate) G&S-Soft, Jakobstr. 4, 6100 Darmstadt 591009 + 711156

Blitter-TOS + altes TOS gleichzeit. in allen ST; auch einzeln Original-Anleitung K. Ratsch, Herner Str. 127; 4350 Recklinghs

BIETE SOFTWARE

■■■■■■■■■■■■■■■■■■■■
■Atari · Schulmeister ST · Atari■
■ Die komfortable Noten- und ■
■Klassenverwaltung mit 40 Me-■
■ nüfunktionen. (GEM) Indivi- ■
■ duelle Programmanpassung ■
■durch editierbare Masken und■
■ Parameter: (Fächer, Unterbe-■
■reiche, Gewichtung, Notentyp,■
■Sortierkriterien ect.) Optimie-■
■rende Druckerausgaben (Epson)■
■500 Kbyte Ram / Schwarzweiß-■
■Monitor. Version für BRD und■
■**Schweiz**. Ausführl. Info/Frei-■
■umschlag. M. Heber-Knobloch■
■Auf der Stelle 27, 7032 Sindelf.■
■■■■■■■■■■■■■■■■■■■■

Von Lehrer für Lehrer! Noten-verw.prg **Markbook plus** in GEM. Über 20 versch. Funkt. f. alle gäng. Notensyst. Spez. vers. auf Wunsch; Info gegen Freiumschlag. H. J. Merkel/Nahestr. 28/6600 SBR

Wärmebedarfsberechnung DIN4701
KZahlberechnung DIN4108
Wärmeschutznachweis WäSchV
Dipl. Ing. V. Koch, Am Mehnacker 11 3563 Dautphetal 3, Tel. 064687652

■■■!!PD jetzt DM 5,50!!■■■
■PD von 1 – diese ST-Ausgabe,■
■auf **Mark.-Disk SS je DM 5,50,**■
■P&V. DM 5,00, ab 20 St. frei!■
■V.-Scheck od. NN (+ DM 1,70)■
■■ NEU! ■ NEU! ■ NEU! ■■
■ **Weg mit dem PD-Müll –** ■
■ **Qualität vor Quantität!!!** ■
■**Jetzt eigene PD-Sammlung** m.■
■ **Bereichssortierung, als Disk** ■
■**o. einzel. PRG's** nach Kbyte!■
■ **Viele Sonderangebote!!!** ■
■2,50 DM-PD-Disk, Gratis-PD..!■
■ **Super Gratis-Katalog!!!** ■
■J. Rangnow · 7519 Eppingen■
■Talstr. 8 · 07262/5131 ab 17°°■
■■■ **PD-Express-Versand** ■■■

★ ★ ★ **PD-Service** ★ ★ ★
Angebot wie PD dieser Ausgabe
Einzeldiskette (ab 6) **DM 5,00**
bis 5 Stück **DM 6,00**
Doppeldiskette DS z. B. 1/2...91/92 usw. (aufeinanderf. u. ungerade beginnend **DM 7,00**
Pakete 1–10, 11–20 usw.
auf je 5 Disketten DS **DM 30,00**
Preis inkl. Disk zzgl. Porto u. Verp. DM 5,00 (Ausl. DM 10,00) NN plus 1,70 (besser V-Scheck) ausf. Liste gegen frankierten (1,10) u. adr. Rückumschlag
I K S , Schönblickstr. 7
7516 Karlsbad-Mu
ab 18 Uhr 07202/6793

■ ■ **Suchen Sie PD-Soft?**
■ ■ Interessieren Sie sich
■■ für etwas neues??
■ ■ Gratiskatalog anfordern!
■ ■ **Klaus Kohler**
Don-Carlos-Str. 33B
7 Stuttgart 80

Public-Domain-Blitzversand!
Riesenauswahl! Preise:
ab DM 4,– einseitige Disketten
ab DM 6,– doppelseitige
Inclusive Diskette!!!
Auch alle ST-Disks! Auf Doppel-disks beliebig kombinierbar!
Gratisliste anfordern bei:
A. Gauger Software
Buhlstraße 16a, 7505 Ettlingen
0 72 43 / 3 18 28
Neu: Neuerscheinungen an Originalsoftware für den ST liefern wir ab Lager!

Wärmebedarf DIN4701 + K-Zahl ★ Heizflächenauslegung ★ Rohrnetz ★ Demodisk 2-seitig für DM 10 Vorkasse von J. Binder Eichendorffstr. 15 · 5030 Hürth

★ ★ ★ ★ **PD-SERVICE** ★ ★ ★ ★
★**PD-Angebot dieser Ausgabe**★
★ **Einzeldiskette SS DM 5,00** ★
★**Doppeldiskette DS z. B. 1+2**★
★**33+34, 89+90 usw. DM 8,00**★
★ **Pakete 1–10, 11–20, usw.** ★
★ **auf je 10 Disk. DM 40,00** ★
★ **auf je 5 D. (DS) DM 35,00** ★
★**Porto u. Verpack. DM 3,00**★
★**V.-Scheck o. NN (+DM 3,50)**★
★ **E. Twardoch, Kais.-Wilhelm-**★
★ **Str. 88 – 1000 Berlin 46** ★

★ ★ **Public Domain Software** ★ ★
★ Atari ST-Disk ab DM 3,00 ★
★ **Katalog-Diskette gratis** ★
★ Bernd Schick 8542 Roth ★
★ ★ Hans-Breckwoldt-Str. 30 ★

■■■■■■■■■■■■■■■■■■■■
■ ★ Atari ★ Schulmeister ST ■
■ Die komfortable Noten- u. ■
■ Klassenverwaltung mit ca. 40■
■Menüpunkten/GEM. Editier-■
■ bare Masken und Parameter: ■
■ (Fächer, Unterbereiche, Ge- ■
■wichtung, Notentyp, Sortierkri-■
■ terien ect.) 500 Kbyte Ram ■
■monochrom-Monitor. Version■
■für BRD und SCHWEIZ. Aus-■
■ führl. Info/Freiumschlag. ■
■ M. Heber-Knobloch ■
■ Auf der Stelle 27 ■
■ 7030 Sindelfingen ■

Neu! 1987+88 Lohnsteuerjahresausgleich + Est-Berechnung + Lst + Est-Tabellen / Mandantenfähig 30 DM Vork. Höfer Grunewald 2a 5272 Wipperfürth, T. 02192/3368

ST-PD Disketten dieser Ausgabe ab DM 2,– Eprommbrenndienst Info: ST-PD, A. Witaseck, Josef-Neuberger-Str. 26, 4 Düsseldorf

★ Public-Domain-Kopierservice ★
15 MByte NEUSTE Software ★
Alle Disketten dieser Ausgabe ★
Einseitig 5,50 ★ Zweiseitig 6,50 ★
Info kostenlos ★ Tel. 02954-1050

Minerale-Prg. m. Druckerausgabe von Achat bis Zoisit über 250 Minerale. Best. mit Überweisung v. 25,–; Georg Pskto 90203-107 PGA BluW. Abs. nicht vergessen! Prg a. 1DD, Verp.-Versandk. incl.

Daten/Programmverluste durch Domputerviren! Schützen Sie sich! Prg: 40,– DM (Scheck/Bar) Info: Gratis / M. Klocke-Sewing Gronauer Str. 19 · 48 Bielefeld 1

65C02 Entwicklungs-System für ATARI ST. Mit Assembler und Emulator. Perfekt geeignet für Micro-Controller. Entwickler + nur 98 DM. M. Stent, Brendelstr. 2, 6000 Frankfurt 90, T. 069-784442

Beckertext ST Orig., m. Handbuch 9/87, DM 110,– Tel. 0211/636409

KFZ-ST: Erfassung, Berechnung u. graf. Dokumentation aller KFZ-Kosten unter GEM. Jeder Drucker anschließbar. Disk 59 DM, Demo 10 DM. Dipl.-Ing. H. Heydrich, Lange Zeile 84 · 8520 Erlangen

PD-Grafik-Bilder, Art-Lib für Degas. Katalog: Frey, Rheinstr. 12A 6538 Münster-Sarmsheim

KUNO – Kursnotenverwaltung für SEK.II, 65 DM. Info gegen Rückporto: L. Kraft Kastanienweg 7, 6730 Neustadt/W.

Biete **COMPILER** für ST Basic! 60 Befehle. 3–10 mal schneller TOS + GEM Biblio. Inc. Vers+Demo 20 DM. C. Jakfeld 0208/640184

Protext/Beckertext Druckertreiber für Star NL-10 alle Schriften in NLQ IBM/Centr. Intrf. ausführl. Anl. 30 DM inc. Disk; U. Köhler, Mont-Cenis Str. 537, 4690 Herne 1

Freie Tastaturbelegung mit allen Sonderzeichen, DM 30,00. Info gegen Rückumschlag. Helbo, Badestr. 1, 3150 Peine

★**Super-Lohn-Einkommensteuer** Jahresausgleich 87 neu m. Datenspeicher, Kundenverw., Formulardruck, Analyse ab 70,– jährl. Aktual. (10,–). Info gg. RP PC-Demodisk 10,–. H-I-Software H-Ilchen · Niederfelderstraße 44, 8072 Manching · Tel. 08459-1669

Große Grafiksammlung!!! Ca. 2000 IMG-Grafiken (z. B. für Wordplus) DM 20,– 06302-3338

■ **GRATIS PD** bei Zusendung■ einer Leerdisk + Freiumschlag Ronald Baumann, 8371 Kirchdorf

★**Turbo-Spice, Original, 99 DM** ★(NP 140,– DM), Gerhard Huber ★Jayamannstr. 7,. 8318 Bonbruck

ST-PD-Mailbox 0211-719261 8N1

■ **PD-Disk nur 4,00 DM!** ■
■ aus ST-Nr. 1 bis heute ■
■ H. Schneider 0281/61772 ■

★ ★ FIBU-PRG DM 49,– ★ ★ Maus-Bedienung, B/D-Ausgaben 60 Konten / 2500 Buchungen E/Ü-Rechn. (monatl.) GuV/Bilanz MwSt. autom. Tel.: 09420/968

LOGISTIX: Orig. Zust. 250 DM **LATTICE-C** V3.04: Orig., 200 DM Tel. 02514122982

Ganzseitenbildschirm auf SM124 Emulator + 80 Zeilen-Editor f. C, Basic und andere ASCII-Tex. über 40 Funkt. + Macrotasten, 30,– p. Rech., Info 0231/441908

CHIRO-ST, Handlesen, 1M 2S, DM 59,– Tel. 02424/7852

STEVE Text-Grafik-Datenbank Superprogramm – unbenutzt DM 350,– 05462/1808

Original-Software:
Paintworks (Activision) DM 70
The Musik Studio (dto.) DM 50
K-SPREAD (Kuma) DM 70
zusammen nur DM 150
E. Hartwig, Zähr. Allee 84, Minden

Index-Datei-Tool (Megamax-C), Druck-PRG: Etikett, Serienbrief, Liste Tel. 02551/5509 Michael Lanczak, Stettiner Str. 9 4430 Steinfurt

Vokabelprgm VocMaster +, V: 2.5 Grundwortschatz, Synonyms, GEM 49,–, Info: Gratis, Demo: 20,– Reiner Kocher, T: 089/3134946 Caracciolastr. 16, 8 München 45

Kassenbuch + Banküberweisung je 49 DM B. Eichhorn T. 030/667277

Beckertext, 140 DM, 0201/7100425

BS-Handel Vers. 2.02 DM 600,– 07371/7267 od. 1760

Software für ST: 0421/583485!!!

ST-Soft!!!: Call: 0421/583485!!!

PD-Software – Info-Disk 4,- DM (bei Rücksendung Gutschrift!!) D. Metz; Bahnhofstr. 84 A 2153 Neu Wulmstorf

★ ★**PD-Aktuell,** das Infomagazin für den PD-Anwender. Probeheft bei **EUSAX PD** 4018 Langenfeld Haus Gravener Str. 85 T. 81232

PD-Software ab 2,– DM/Diskette Angebot wie PD dieser Ausgabe Tel. 02721/2432

PC-DITTO 120,- DM 0541-596539

★ ★ ★ **C W - C H A R T** ★ ★ ★ Börsensoftware Neu: Version 5.0 Demo 10 DM bei Weisenböhler Waldstr. 2, 7175 Vellbg. 079071309

xxxx **NEC P6-TREIBER** xxxx f. WORD+ spez. f. Ex.-Arbeiten Prop+Block; 1,5zg; 1zg f. Zitate xxxx 0203/55 77 29 xxxx

STAR NL-10 Utility für Atari ST Kursivschrift in NLQ u. Draft: Accessory mit Druckeranpassungen Handbuch; DM 30; 040/868779

OK-FIBU, Buchführung für Freiberufler-Vereine-Privat, G+V, Bilanz, Kostenst., Abschluß, MWSt einf. Bedienung. Nur **69,– DM!**
OK-FAKT, Lager- u. Kundenverwaltung, Rechnung, Serienbrief, Umsatz. Nur **49,– DM!**
OK-HANDEL zus. nur **98,– DM!**
MITGLIEDER-Verwaltung für Vereine u. Sonstige; Serienbrief, Beitrag, Mahnen, Abbuchen, Statistik etc. Nur **69,– DM!!!**
Info: Ottmar Kuschek, Mühlheimer Mühle, 5378 Blankenheim.

Atari 520STM SF354 Maus Joyst. Monitor. 800XL Datasette. Beide + weit. Zubehör u. abs. neuwertig Sehr preisgünst. Tel. 0911/736938

Starwriter ST, Protext, Beckertext, Druckertreiber für STAR NL 10 fLC 10 (neu), über 70 Schriften; 30,– DM; M. Köhler, Mont-Cenis Str. 537, 4690 Herne 1

★ ★ **Dringend zu verkaufen** ★ ★ Wegen Systemumstellung ATARI ST Software preisgünstig abzugeben **Modula-2 TDI Version 3.0 Developer Version – > 190 DM** Tel: 02202/21784 abends

★ ★ **Achtung Handwerker** ★ ★ Kunden, Lager, Auftragsbearbeit. Automatisches Mahnwesen, Serienbriefe, 1STWord Kompatibel! Preis: DM 249 – Demo DM 10. T. Berdowski · Tel. 06121-301102 Postfach 1807 · 6200 Wiesbaden

Vorsicht! WMS(376237) kassiert zwar immer im Voraus, liefert aber oft defekt, mitunter gar nicht! Habe Lehrgeld bezahlt! Näheres 06151/62321

■**Megamax C 260,- 089/988796**■

Von Bauleiter für **Bauleiter!!** **Aufmass-Massen-Abrechnungs-Prg** f. Atari-ST 245,- T. 0821-499518

Privatliquidation / Buchhaltung Rechnungen n. GOÄ'82, Mahnwesen Einnahme-/Überschußrechnung für Steuererklärung; besonders für nebenamtlich selbständige Ärzte geeignet. **Tel.: (02247) 4937**

James Seite 51

SUCHE HARDWARE

1040 STF + Monitor 0251/216795

James Seite 51

KONTAKTE

Midi + Homerecording-User-Club nimmt wieder Mitglieder auf!! Info gegen Rückporto von: Kay-Uwe Berghof Roseggerstraße 5 5600 Wuppertal 2

★ ★**ST-Softwareentwicklung**★ ★ Schreibe Software, die speziell auf Ihre Bedürfnisse zugeschnitten ist. Ksl. Info anford. Thomas Leufkes, Knappenstr. 10, 4353 Oer-Erkenschwick, 02368/55020

Achtung **Händler!** Wir suchen für unsere **Branchenlösung** für **Immobilienmakler Vertriebspartner** in der **BRD, Österreich** und der **Schweiz.** Anfragen (mit Händlernachweis) an: Kappert Computersysteme Postfach 1269, D-8980 Oberstdorf

James Seite 51

SUCHE SOFTWARE

HILFE! Keinen Mucks macht mein Drucker, wenn ich Grafik vom Art Direktor ausdrucken lassen will. Konfiguration: 1040 STF, SC1224, NEC PC6 Colour. T. 652533 K. Koslowski, Kolumbusstr. 34 8 Mü 90

PC ditto / habe z. B. VIP Telf. 07552/5332

Pascal, C-Comp., Assembl. Orig. Nils Faltin Tel. 09126/8840

Suche Programm für Vereinsverwaltung. 06831/707061 Ferring

James Seite 51

VERSCHIEDENES

★ NEC-1037 Diskstationen ★
★ Einzel- und Doppellaufw. ★
★ Vortex Festpl. Eprom-Disk ★
★ 128k. Billiger als anderswo! ★
★ Weide Uhr-TOS. ★
★ Verhandelt mit uns ★
★ Softw. billig – Liste! ★
★STEVE, SIGNUM, STAD u.a.★
★ ATARIPOINT 07031/52607 ★

SCHREIBERLING gesucht, der mir Wissenschaftl. Text auf 1. Word plus od. Signum schreibt. 089-6379776

Platinen-Layouts! von Elektronik-sch., Atari-ST **Plotterservice** f. alle CAD Progr., 0431-180975 U. Tams, Westring 273, 2300 Kiel

ST 86+87 zu verkaufen nur Kpl. DM 130,– 07243/14137 ab 19 Uhr

CH: Unterstützung in allen prof. ST-Anwendungen (Datenbank, Kalkulation, Textverarb., Fibu); Beratung, Einführung sowie Installation von ganzen Systemen. Tel. CH 043 31 28 31

Übernehme Programmieraufgaben in GFA BASIC P. Neurohr, Eschersheimer-Landstr. 273, Tel: 069/57298

Betr.: Noch mehr Festplatten Utilities ST 2/88

Als Hersteller des Cachespeichers 1ST Speeder waren wir über Ihren "Testbericht" sehr verärgert. Er ist nicht nur schlecht geschrieben und miserabel recherciert - er enthält auch schlicht fehlerhafte Angaben.

Beim Anlesen dieses "Tests" fiel von Anfang an auf, daß offensichtlich durch die Sprachwahl des Berichts auf eine Herabwürdigung des Produkts gezielt wurde. Wir können uns dieses Eindruckes gerade deshalb nicht verwehren, da an diversen Stellen (z.B. ist 1ST Speeder offensichtlich in C oder in sehr ungeschicktem Assembler geschrieben) anscheinend auf billige Polemik anstat auf Argumente Wert gelegt wird.

Geradezu unverschämt fanden wir den Satz: "Hoffentlich verwendet der Hersteller nicht ausgerechnet den letzten Satz, um für sein Produkt zu werben". Hier wird beabsichtigt, ein schiefes Bild zu Erzeugen, in dem wir bewußt, ohne jeglichen Anlaß herabgesetzt werden. Normalerweise ist dies ein Stil, in dem sich Boulevardblätter bewegen - er sollte einer Zeitschrift wie ST unwürdig sein.

Weiterhin bleibt festzustellen, daß der Autor des Berichtes nach unseren Unterlagen eine sehr alte Version getestet hatte, die schon länger durch eine neuere, noch schnellere abgelöst worden ist. Ihr Autor befand es nicht für nötig, vor Abschluß des Tests bei uns nachzufragen, ob eine neue Version ausgeliefert würde bzw. welche Verbesserungen geplant seien.

An dieser Stelle möchten wir noch auf folgende Dinge hinweisen: Nach Erfahrungen werden Cachespeicher zu etwa 85% für größere Datenmengen (>50 KB) eingesetzt. Der alte 1ST Speeder wurde dahingehend konzipiert. Ihr Autor CB ignorierte in seiner Gesamtbeurteilung allerdings seine eigene Erkenntnis, daß wir ab etwa dieser Größe wesentlich schneller waren, als unsere Konkurrenzprodukte. Genauso widerspricht er sich wenn er zum einen von einem "problemlosen Festplattenbetrieb" und zum anderen über Probleme mit der Festplatte schreibt weil "man sich ja drum kümmern muß, daß sich Festplatte und Cache vertagen". Die Probleme mit mediachange sind bekannt - Sie sind in der neuesten Version behoben. Bessere Recherce tut hier dringend Not!

Abschließend hinterläßt bei uns (oder bei unseren Kunden: zu folgern aus unseren Kundenbriefen) weniger 1ST Speeder ein "höchst zwiespältiges Gefühl", sondern eher die Machart dieses tendenziösen Artikels.

TOMMY SOFTWARE,
Thomas Maier, Frankfurt/M.

Werter Herr Maier!

Gehen wir Ihre Vorwürfe einzeln an:

1. "in C (oder in sehr ungeschicktem Assembler) geschrieben..."

Daß der Speeder schlecht programmiert wäre, habe ich nie behauptet. Natürlich kann man in einen Satz beliebig viel hineindeuten; das weiß jeder, der schon unter gymnasialen Deutschlehrern gelitten hat. (Protest der Lehrerverbände, Leserflucht und Auflagenschwund, Bankrott der MERLIN.)

Lassen Sie mich den Artikelteil komplett zitieren: "Bei M-CACHE konnte ich das disassemblierte Programmlisting noch leicht von Hand analysieren" - M-CACHE ist in übersichtlichem Assembler geschrieben - "Beim 1stSpeeder ist es schon nicht mehr so einfach, offensichtlich sind größere Programmteile in C geschrieben" - was dazu führt, daß der compilierte Code umständlich wird - "(oder in sehr ungeschicktem Assembler)" - womit ich sagen will: Um bei Assemblerprogrammierung einen C-Compiler-ähnlichen Code zu produzieren, müßte man schon ziemlich "C-aputt" daherhacken. Es liegt mir also fern, Ihre Programmierer zu diffamieren (C-Code sieht nun mal so aus); im Gegenteil: Bezeichne ich nicht die Speeder-Programmierer als "noch cleverer" ob ihrer "Programmierschmankerln" bei den " schnellen Verschieberoutinen" und ob der "effizienten Suche im Puffer"?

2. "Hoffentlich verwendet der Hersteller nicht ausgerechnet..."

Unverfroren, unverschämt, ungezogen, unerhört? Das Zitat wieder im Ganzen: "Ab 32 KB schließlich griff das Cache-Programm (Speeder ist gemeint) auch bei anderen Benchmarkteilen, die bis zu einhundertmal schneller liefen als ohne Puffer. Hoffentlich verwendet der Hersteller nicht ausgerechnet den letzten Satz, um für sein Produkt zu werben; man muß ... berichtigend dazu sagen, daß diese Benchmarks wie geschaffen für Cache-Programme waren. Insgesamt kann man realistischerweise von einer Steigerung um das 5- bis 10fache ausgehen."

Ich wollte also nur ausdrücklich klarstellen, daß die 100fache Steigerung wirklich eine Ausnahme ist. Daß Sie selbst in Ihrer Werbung für den Speeder nur eine 10fache Steigerung reklamieren, ist korrekt und lobenswert; ich möchte Ihnen im Namen vieler Computeranwender danken, die schon auf unlautere Geschwindigkeitsangaben und entstellende Testzitate in der Computerwerbung (anderer Hersteller) hereingefallen sind. Genau auf derartige Werbepolitik im allgemeinen wollte ich ironisierend hinweisen. Sie brauchen den bewußten Satz also nicht als bösartige Unterstellung zu verstehen.

3. "...befand es nicht für nötig, vor Abschluß des Tests bei uns nachzufragen..."

Ja, vielleicht hätte ich nachfragen sollen, als sich - gerade im Floppybetrieb - die vielen Merkwürdigkeiten (um nicht das häßliche Wort "Fehler" zu verwenden) einstellten. Allerdings wäre der Artikel kaum deswegen (noch) positiver ausgefallen, nur weil ich dann hätte schreiben können: "Laut Hersteller werden die Fehler in der nächsten Version behoben sein." Erstens wird mir das jeder sagen; zweitens ist das eine Selbstverständlichkeit. Schließlich hängen von der Funktion des Speeders überall in Computer-Deutschland wichtige Daten ab. Man denke nur an das elektronische Adreßbuch von Herbert aus Remscheid (Name aus Klanggründen geändert...). Drittens können Sie sich auch nicht damit herausreden, es gäbe längst eine neue Version. Die getestete Version wurde jedenfalls ganz offensichtlich längere Zeit so verkauft; sie war ja nicht als Beta-Version gekennzeichnet. Erwarten Sie dafür Lob, daß Sie ein fehlerhaftes Produkt verkauft haben?

4. "... werden Cachespeicher zu etwa 85% für größere Datenmengen (>50KB) eingesetzt."

Ich habe es keinesfalls ignoriert, daß der Speeder bei größeren Puffern blitzartig schnell wurde, auch nicht in der Gesamtbeurteilung. Spreche ich da etwa nicht von "phänomenalen Zeiten" bei großen Pufferspeichern? Ich bestreite nur Ihre These, daß man mit einem Cachespeicher quasi eine RAM-Disk ersetzen kann und deswegen auch ruhig mal 100 KB oder mehr dafür opfern kann.

Vergleichen wir doch RAM-Disk und Cachespeicher bei einer typischen Anwendung: Programmentwicklung per Editor, Compiler, Linker et cetera.

Die einzelnen Programme werden angeklickt und dabei von der Diskette in Puffer- und Anwenderspeicher geladen. Soweit genauso wie bei der RAM-Disk, die ja auch erst gefüllt sein will. Beim Entwickeln des Programms braucht man mit Cache aber wesentlich länger, weil alle Schreibzugriffe auch auf Diskette/Platte durchgereicht werden. Zugegeben, das ist auch einen Hauch sicherer als eine resetfeste RAM-Disk, aber welchen Preis (Zeit = Geld) bezahlt man dafür?

"Wo hab' ich denn jetzt dieses blöde Includefile?" - Diskettenwechsel, der Cachespeicher vergißt alle gepufferten Sektoren der alten Diskette (wenn er richtig funktioniert). Die alte Diskette wird wieder eingelegt, und das Compilerpaket muß komplett von der Diskette zurückgeladen werden (gähn...). Der RAM-Disk-Anwender quittiert das mit einem leichten Zucken in den Mundwinkeln.

"Dann benutzen Sie doch RAM-Disk und Cachespeicher", sagen Sie vielleicht. Also gut: 110 KB für Pufferspeicher und Verwaltung, 500 KB RAM-Disk für den Compiler, 300 KB Wordplus, 50 KB Texte, 70 KB System (Video-RAM, Systemvariablen), 10 KB für Festplattentreiber, 50 KB GEMDOS-Speicherverschnitt, 50 KB Accessories; macht 1150 KB in einem Megabyterechner. "Geht doch gar nicht!", sagen Sie. Eben.

Nein, nein, nein, Cachespeicher sind vornehmlich dazu da, um Zugriffe auf einzelne Sektoren zu beschleunigen (FAT- und Directorysuche). Daß dafür auch 8 KB reichen, zeigt der Konkurrent M-CACHE.

geht weiter...

5. "Genauso widerspricht er sich..."
Sie haben recht, ich schreibe in einem Satz, daß der Festplattenbetrieb problemlos gewesen sei und komme dann am Ende doch noch auf einen Fehler zu sprechen. Es müßte natürlich heißen: "fast alles problemlos". Ich bitte um Entschuldigung, der Satz ist mir bei der Korrektur entgangen.
6. "miserabel recherchiert..."
Ein oberflächlicher Test? Welche Zeitung veröffentlicht 2 1/3 Seiten über ein kleines Cache-Utility? Wer macht sich sonst die Mühe, eigene Benchmark-Programme zu schreiben, die detaillierte Rückschlüsse auf den Algorithmus erlauben? Welcher Autor kniet sich wie ich in den Programmtext hinein, disassembliert und analysiert ihn, um Schwachstellen und Glanzpunkte zu finden?
Noch einmal will ich (wie im Artikel) betonen, daß der Speeder das Zeug zu einem tierisch guten Cache-Programm hat, wenn die Fehler entfernt werden, der Diskettenwechsel 100%ig erkannt wird (was gar nicht so einfach sein dürfte) und man außerdem davon Abstand nimmt, mit dem Speeder eine RAM-Disk ersetzen zu wollen (dann kann man nämlich den Algorithmus so umschreiben, daß auch bei kleinen Puffergrößen die Geschwindigkeit des Speeders zur Geltung kommt). Ich werde mich neuerer Versionen gerne wohlwollend annehmen.
Keep hacking, CB

Betr.: Listing Every/After aus ST 12/87

Ich möchte eine Korrektur aufgrund eines Leserbriefes vornehmen. Ein Leser beschrieb einen Fehler, den ich mir erst auch nicht erklären konnte. Dann habe ich mir aber noch einmal das abgedruckte Listing (Every/After) angesehen und bemerkt, daß ich beim Einfügen von Kommentaren in das Listing, das ich Ihnen geschickt hatte, wohl etwas zu eifrig war. Ich habe dabei versehentlich einen Ausdruck gelöscht, das aber noch nicht bemerkt, obwohl es so in vielen Fällen (Interpreter und ROM-TOS) zu Bomben-Fehlern kommt.
Aus diesem Grund möchte ich Sie bitten, möglichst bald eine Korrektur abzudrucken und so weiteren Briefen vorzubeugen. Diese Korrektur könnte folgendermaßen aussehen:
Korrektur zum Artikel Every/After, Heft 12/87, S.95 ff.:
Auf Seite 95 ist in Zeile 43 leider ein Teil der Zeile versehntlich gelöscht worden. Die Zeile muß wie folgt lauten, da es sonst zu Bomben-Fehlern kommt:

```
If Lpeek(Lpeek(&H456)+4*I%)>Basepage
And
Lpeek(Lpeek(&H456)+4*I%)<=Basepage+
Lpeek(Basepage+12)
```

Lutz Preßler, Hatten

Leser -> ST

Betr.: Update für K-Resource

Wie bereits in Ihrer Zeitschrift erwähnt, läuft K-Resource von Kuma nicht auf dem Mega ST2, auf den ich inzwichen "umgestiegen" bin. Gibt es inzwischen ein Update für dieses Programm, so daß es auf dem Mega ST2 einsatzfähig ist?
Peter Halter, Heidelberg

Red.: Die Antwort könnte von Radio Eriwan kommen und lautet folglich: Im Prinzip ja! K-Resource gibt es in einer Mega-tauglichen Version 1.1 von Application Systems /// Heidelberg, wird aber nur in Verbindung zu einem Produkt (Megamax C oder Megamax Modula 2) dieser Firma gegen ca. DM 50,- ausgeliefert. Dies ist die einzige derzeit auf den Mega STs laufende Version. Nähere Fragen unter der Telefonnummer 06221/300002.

Betr.: Betriebssystemroutine $4B (Pexec)

Ich besitze seit einiger Zeit einen ATARI ST und programmiere auch in Maschinensprache. Nun habe ich folgendes Problem: Wenn ich von Maschinensprache aus mit der Betriebssystemroutine $4B ein anderes Programm nachladen will, so habe ich bei Programmen, welche die Maus benutzen, Schwierigkeiten. Diese Programmen werden zwar tadellos geladen und laufen auch ohne Probleme, jedoch ist der Mauszeiger unsichtbar. Vorhanden ist er jedoch, was man daran sieht, daß die Pulldown-Menüs runterklappen, wenn der unsichtbare Mauszeiger sie berührt. Offentsichtlich mache ich etwas falsch! Nur was? Vielleicht können Sie mir einen Tip geben.
Axel Kowald, Berlin

Red.: Die Erklärung dafür dürfte darin liegen, daß Sie in Ihrem Programm, von dem Sie das zweite Programm mit Pexec aufrufen, die Maus nicht eingeschaltet haben. Mit der AES-Routine GRAF_MOUSE (78) und dem Parameter gr_monumber=257 ist dies leicht zu bewerkstelligen.

Betr.: Probleme mit 1ST Word Plus und Etikettendruck.

Ich versuche schon seit geraumer Zeit meinem Star NL10 das Etikettendrucken beizubringen. Z.B. 100 Etioketten mit meiner Adresse. Nachdem ich im Auswahlmenü die Anzahl angegeben und auf NLQ geklickt habe, beginnt er auch brav den Druckvorgang. Doch nach dem ersten Etikett beginnt dann erst einmal ein atemberaubender Papierausstoß, ohne daß gedruckt wird. Da sich mein Englisch in Grenzen hält, konnte mir das Handbuch auch noch nicht viel weiterhelfen. Als Druckertreiber benutze ich übrigens den PD-Druckertreiber aus der ST.
Hanno Sekuterski, Arnsberg

Red.: Das mit dem Papiervorschub ist normal. Ihr 1st Word schickt nach jeder Seite einen Formfeed. Beeinflussen könnte man das durch Änderung des Druckertreibers, aber das ist nicht gerade der geniale Weg.
Eine Möglichkeit ist folgende: Schreiben Sie die Ganze Seite mit Ihrer Adresse voll und stellen Sie im Layout-Menü die Anzahl der Zeilen auf 72.
Wesentlich eleganter geht der Adressendruck aber mit dem Programm ADR.2 aus der PD-Sammlung (Disk 93). Es macht nebenbei noch eine komplette Adreßverwaltung, Listendruck und Serienbriefe mit 1st Word oder mit anderen Testprogammen.

Betr.: Blitter-TOS auf Diskette

Seit einiger Zeit gibt es endlich das neue Blitter-TOS. Da ich noch einen älteren 520+ mit RAM-TOS habe, wäre es für mich sehr schön, wenn Sie dieses neue TOS (wie seinerzeit die ROM-Version) in ihrem PD-Service anbieten würden. Denn es ist für sicherlich kein Problen dieses neue TOS zu booten. Interressant ist dies deshalb, weil doch einige - wenn auch nicht alle - Fehler des alten TOS (... "_" läßt grüßen) entfernt wurden. Ich hoffe nur, daß ATARI das Blitter-TOS auch auf Diskette herausgibt.
Martin Josef Warken, Sankt Wendel

Red.: Leider gibt es zur Zeit keine Diskettenversion des Blitter-TOS; laut letzter Auskunft von ATARI ist noch nicht einmal eine Version geplant. Dies ist wohl dadurch begründet, daß ATARI lieber seine ROM-Version, die ja einzeln erhältlich ist, verkauft. Somit besteht für uns keine Möglichkeit eine Diskettenversion anzubieten, da wir dazu die Genehmigung von ATARI (wegen Copyright) benötigten.

Leser -> Leser

Ich habe folgendes Problem: Ich arbeite mit dem Makroassembler von GST und ST Pascal plus von CCD. Wenn ich aber Programmteile in Pascal mit Teilen in Assembler zusammenlinken möchte, so ist das nicht möglich; die Formate der Objektfiles sind verschieden.
Wer weiß also, wie man dieses beheben kann oder wie die ".O"-Files des ST Pascal plus, die übrigens zu denen des Entwicklungspaketes sein sollen, aufgebaut sind?
GST hat sein Format nämlich im Handbuch beschrieben, während die Firma CCD auf Anfrage nicht berreit war, mir ihres zu erläutern, obwohl ich eingetragener Benutzer, d.h. Käufer und nicht Raubkopierer bin.
Michael Naatz, Rudolf-Kinau-Str. 28, 2080 Pinnenberg



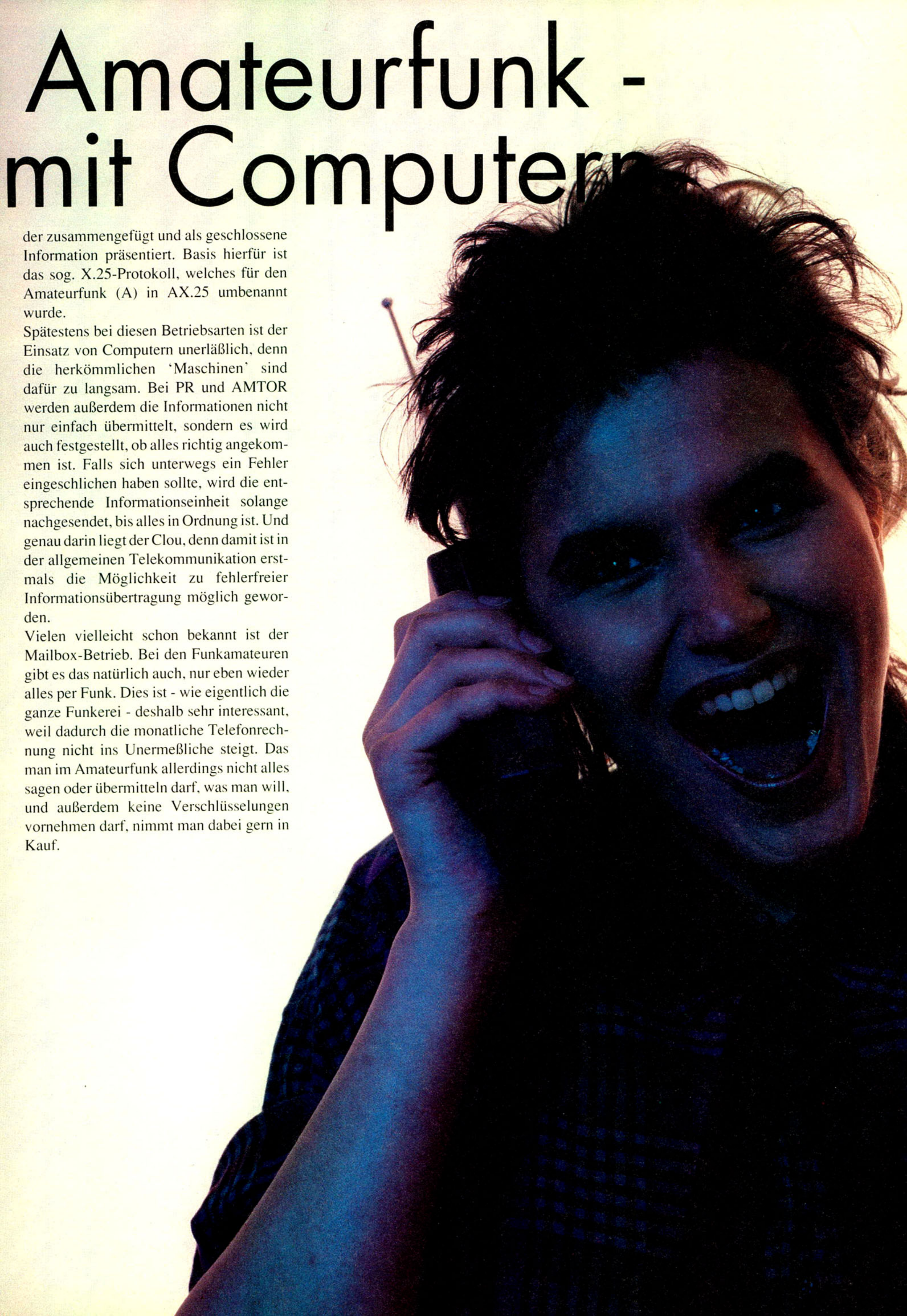
Computer im Amateurfunk

1. Teil - eine allgemeine Einführung

Was ein Computer ist, braucht an dieser Stelle wohl weniger diskutiert zu werden. Was Amateurfunk bzw. ein Funkamateur ist, wahrscheinlich schon eher. Hier beginnen erfahrungsgemäß nämlich die ersten Verwechslungen mit dem CB-Funk, dem sog. Jedermann-Funk.

Der Hauptunterschied liegt darin, daß ein Funkamateur - von denen es in Deutschland rund 60.000 gibt - im Gegensatz zu einem CB-Funker, eine umfangreiche Prüfung bei der Deutschen Bundespost abgelegt und dafür die Erlaubnis erteilt bekommen hat, in 22 statt in 2 Frequenzbereichen zwischen 1.815 kHz und 250 GHz (!) - allerdings abhängig von der Lizenzklasse - funken zu dürfen.
Der zweite wesentliche Unterschied liegt in den unterschiedlichen Übermittlungsverfahren, im Fachjargon Betriebsarten genannt. Im Gegensatz zum CB-Funker, der auf Sprachkommunikation in Amplitudenmodulation (AM) oder Frequenzmodulation (FM) beschränkt ist, darf hier der Funkamateur sehr viel mehr.
Das beginnt mit den klassischen Morsezeichen, im Fachjargon kurz CW oder auch Continuous Wave genannt. Dieser Betriebsart sehr ähnlich ist das Funkfernschreiben (RTTY = Radio-Tele-TYping)). Was ein Fernschreiber ist, wissen wohl die meisten. Das Ganze muß man sich nun einfach per Funk vorstellen.
Hier wird jedes Zeichen in einem 5-Bit-Kode dargestellt. Da man mit einem solchen Kode allerdings nur 2 hoch 5 = 32 unterschiedliche Zeichen darstellen kann, aber für das Alphabet plus Zahlen und Sonderzeichen wesentlich mehr benötigt, wird hier ein Trick angewendet: man benutzt den Kode doppelt und sagt vorher, ob es sich um einen Buchstaben oder um eine Zahl bzw. ein Sonderzeichen handelt.
Im Gegensatz zu CW kann man RTTY nicht mehr manuell abwickeln. Hier beginnt entweder der Einsatz eines klassischen Fernschreibers oder eben eines Computers. Der große Vorteil des Computers liegt vor allem darin, daß er keinen Krach macht. Das weiß allerdings nur der zu schätzen, der mal neben einem Fernschreiber gearbeitet hat.
Da sowohl CW als auch RTTY fehleranfällige Betriebsarten sind, hat man sich etwas Neues einfallen lassen: zum einen das sog. Amateur-Microprocessor-Teleprinter-Over-Radio (AMTOR-) und zum anderen das Packet-Radio-Verfahren, kurz PR genannt. Im Amateurfunk läuft übrigens alles mit internationalen, meist englischen Abkürzungen, was den Vorteil hat, daß man sich auch ohne große Sprachkenntnisse erstaunlich gut unterhalten kann.
Wer schon mal etwas vom ISDN-Netz der Deutschen Bundespost gehört hat, der weiß vielleicht, daß dort künftig alles digital ablaufen soll. Basis dafür ist auch dort ein sog. Paket-Übermittlungsverfahren, bei dem jede Informationseinheit (=Paket oder Packet) eine Empfangs- und eine Absenderadresse erhält. Auf diese Weise kann sich diese Informationseinheit ihren Weg selbständig zum Empfänger suchen. Dort werden die ankommenden Einheiten wie-

Amateurfunk - mit Computern



der zusammengefügt und als geschlossene Information präsentiert. Basis hierfür ist das sog. X.25-Protokoll, welches für den Amateurfunk (A) in AX.25 umbenannt wurde.

Spätestens bei diesen Betriebsarten ist der Einsatz von Computern unerläßlich, denn die herkömmlichen 'Maschinen' sind dafür zu langsam. Bei PR und AMTOR werden außerdem die Informationen nicht nur einfach übermittelt, sondern es wird auch festgestellt, ob alles richtig angekommen ist. Falls sich unterwegs ein Fehler eingeschlichen haben sollte, wird die entsprechende Informationseinheit solange nachgesendet, bis alles in Ordnung ist. Und genau darin liegt der Clou, denn damit ist in der allgemeinen Telekommunikation erstmals die Möglichkeit zu fehlerfreier Informationsübertragung möglich geworden.

Vielen vielleicht schon bekannt ist der Mailbox-Betrieb. Bei den Funkamateuren gibt es das natürlich auch, nur eben wieder alles per Funk. Dies ist - wie eigentlich die ganze Funkerei - deshalb sehr interessant, weil dadurch die monatliche Telefonrechnung nicht ins Unermeßliche steigt. Das man im Amateurfunk allerdings nicht alles sagen oder übermitteln darf, was man will, und außerdem keine Verschlüsselungen vornehmen darf, nimmt man dabei gern in Kauf.

Ein weiteres interessantes Gebiet ist der Satellitenfunk. Sie haben richtig gelesen. Die Funkamateure haben über ein Dutzend Satelliten im Weltraum kreisen. Hier wird es nun ohne Computer schwierig, ständig den aktuellen Standort eines Satelliten zu berechnen, um die Antennen entsprechend nachzuführen, und zwar gleichzeitig sowohl in horizontaler als auch in vertikaler Richtung.

Ein entsprechendes Programm kann einem dazu auch gleich noch zeigen, welchen Erdteil der Satellit zur Zeit abdeckt, so daß der Funker weiß, welches Land er über diesen Satelliten erreichen kann.

Es gibt übrigens einen 'Satelliten', der schon sehr alt ist - den Mond. Der Mond ist natürlich kein echter Satellit im technischen Sinn. Einige Spezialisten benutzen ihn als Reflektor, um so Erdteile zu erreichen, die auf den hohen UKW-Frequenzen nicht zu erreichen wären. Hierzu ist natürlich die jeweilige Mondposition wichtig, damit die Antenne genau ausgerichtet bzw. nachgeführt werden kann. Manuelle Berechnungen würden schon überholt sein, bevor sie fertig sind. Für das Funken bliebe dann überhaupt keine Zeit mehr.

Wie man sieht, gibt es heute auch im Amateurfunk Dinge, die ohne Computer undenkbar wären. Allerdings gibt es auch viele Anwendungen, die durch den Einsatz eines Computers lediglich leichter werden. Diese Möglichkeiten werden im folgenden näher beschrieben.

In vielen Ländern auf der Welt gibt es sog. Relais und Baken. Die Relais dienen dazu, Verbindungen zu ermöglichen, die sonst nicht möglich wären, ähnlich wie bei den Satelliten, nur im kleineren Maßstab.

Dazu ist es wichtig zu wissen, daß diese Relais im UKW-Bereich arbeiten. In diesen Bereichen breiten sich die Funkwellen quasi-optisch aus. D.h. im Prinzip, wohin man nicht sehen kann, kann man auch nicht hinfunken. Dieser Umstand ist vor allem für die Autofahrer besonders von Nachteil, da sie sich oft in ungünstigen Gebieten oder zwischen den Hochhäusern einer Stadt aufhalten. Hier helfen nun die Relais.

Relais stehen an exponierten Standorten, so z.B. auf einem Hochhaus oder Berg. Damit sind sie von jeder Stelle der Umgebung zu erreichen. Solche Relais arbeiten vollautomatisch. Sie strahlen das empfangene Signal im selben Moment auf einer anderen Frequenz wieder aus. Daduch können sich zwei Funker unterhalten, die sich sonst nicht hören würden.

Im Gegensatz zu den Relais arbeiten die Baken sehr einseitig. Sie geben wie ein Leuchtturm im Wasser lediglich in bestimmten Zeitabständen ein Signal ab. Damit können die sog. Ausbreitungsbedingungen für die Funkwellen festgestellt werden, denn in Abhängigkeit von bestimmten Wetterlagen gibt es Situationen, in denen Entfernungen überbrückt werden können, die unter normalen Umständen nicht überwunden werden können. Im Fernsehen macht sich das oft durch verzerrte Bilder oder Wellenlinien bemerkbar und manchmal wird auch dort dann von sog. Überreichweiten gesprochen.

Da es in Deutschland und Europa sehr viele solcher Relais und Baken gibt, ist es sinnvoll, hierüber ein Verzeichnis auf Basis einer Datenbank anzulegen. Damit können sehr schnell der Standort, die Frequenz sowie die Betriebsart festgestellt werden. Die oben bereits erwähnten Mailboxen lassen sich über eine solche Datenbank natürlich ebenfalls verwalten und abfragen.

Im Amateurfunk ist es üblich, sich für jede Erstverbindung eine Bestätigungskarte zu schicken. Dies ist eine sehr schöne Sitte, die allerdings auch mit Arbeit verbunden ist. So muß zunächst mal die Karte ausgefüllt und schließlich auch festgestellt werden, ob man diesem Funkfreund schon eine solche Karte geschickt hat oder von ihm schon eine erhalten hat. Nach Ansammlung von einigen tausend Karten kann das zu einer sehr langwierigen Sucherei werden.

Das auch hierfür ein Computer mit einer entsprechenden Datenbank eine große Hilfe darstellt, dürfte jedem klar sein. Doch nicht nur die Verwaltung dieser Karten ist interessant, sondern auch eine Abfrage der Datenbank, welche Verbindungen auf welchen Frequenzen in welcher Betriebsart und mit welchen Ländern getätigt wurden. Ab einer bestimmten Anzahl von Verbindungen aufgrund der unterschiedlichsten Kriterien kann man dann für seine Mühe die tollsten Diplome erhalten.

Doch nicht nur die Erstverbindung, sondern auch jede weitere Verbindung mit dem gleichen Funkfreund muß vom Funkamateur laut Vorschrift der Post aufgezeichnet werden. Auch diese Aufzeichnungen können über einen Computer abgewickelt werden. Daneben gibt es noch eine ganze Reihe von Listen wie Frequenzpläne, best. Landeskennzeichen oder Relaispläne, die sich sinnvoll im Computer abspeichern lassen, um sie jederzeit schnell zur Verfügung zu haben. So ist es auch interessant zu wissen, in welche Länder man seine Funkstation ohne besondere Genehmigung mitnehmen darf.

Funkamateure sind Aktivisten, und so gibt es die verschiedensten Funkwettbewerbe. Auch hierfür lohnt es sich, entsprechende Programme zu schreiben, die einem die Verwaltungsarbeit abnehmen, denn die Auswertung eines solchen Wettbewerbes ist nicht immer ganz einfach. Oft ist es üblich, jeden überbrückten Kilometer mit jeweils einem Punkt zu belohnen. Doch wie weit ist es nun von einem bestimmten Standort in Nürnberg zu einem bestimmten Standort in Hamburg? Das Dumme an der Sache ist vor allem, das die Erde auch noch rund ist, und damit die Berechnung erschwert.

Außerdem ist es während des Funkbetriebes wichtig zu wissen, in welcher Richtung der jeweilige Funkpartner sitzt, um die Antenne in die richtige Richtung zu stellen. Wer würde seine Antenne schon Richtung Norden stellen, wenn er z.B. nach Alaska funken will? Die herkömmlichen Landkarten führen einen hier völlig in die Irre.

Zu Beginn dieses Artikels wurde von den Morsezeichen gesprochen. Leider ist das eine 'Sprache', die man nicht in der Schule lernt, so daß hier sozusagen eine neue Sprache erlernt werden muß. Auch hierbei hilft einem der Computer, vor allem sein Zufallsgenerator. Wer allerdings einen Funkfreund hat, der das Ganze schon beherrscht und immer Zeit hat, mit einem zu üben, der benötigt diesen Programmteil nicht.

Falls Sie nach dem Lesen dieses Artikels etwas neugierig geworden sind, dann sollten Sie sich auch den zweiten Teil anschauen. Dort wird ein integriertes Amateurfunkprogramm für den ST vorgestellt, welches all die hier beschriebenen Punkte und noch vieles mehr in sich vereint.

Der Vollständigkeit halber sei erwähnt, daß es auch verschiedene Arten von Amateurfunkfernsehen und den Empfang von Wettersatelliten gibt. Diese Themen, vom erhöhten Aufwand an Hard- und Software einmal abgesehen, werden nicht näher betrachtet.

Auch kann man den Computer für die Berechnungen von Antennen, Schwingkreisen, Netzwerken und für diverse andere technische Berechnungen einsetzen. Hier gibt es mit Sicherheit kaum Grenzen.

Das im 2. Teil beschriebene Programm konzentriert sich auf all jene Berechnungen und Hilfsmittel, die ein Funkamateur im allgemeinen während einer Funkverbindung und nicht während des Bastelns benötigt.

Wolfgang Cramer

ENDE

OMIKRON.BASIC
Das Buch zum Handbuch

BESUCHEN SIE UNS
HANNOVER MESSE
CeBIT'88
16. – 23. MÄRZ 1988
HALLE17 STAND A70

Aus dem Inhalt dieses Buches:

Für den absoluten Neuling:

Kurze Einführung in die BASIC-Programmierung

Über das Handbuch hinausgehende Beschreibung vieler Befehle, Besonderheiten, Tricks und Kniffe





Verwendung selbstdefinierter Prozeduren und Funktionen

Viele Beispiele, Aufgaben mit Lösungen

Für den Aufsteiger, aber auch für den geneigten Anfänger:

OMIKRON.Sprites - Tücken, Vorteile, Anwendung

Overlay-Technik (Auslagern langer Programmteile und Laden bei Gebrauch)

Grundlagen der strukturierten Programmierung

Schreiben eigener und Verwenden fremder Libraries (Bibliotheks-Funktionen)

Aufrufe und Nutzung von TOS und GEM im BASIC (GEMLib); endlich die Wahrheit über die GEM-Aufrufe! Dabei wird auch das GEM-Zusatzprogramm GDOS berücksichtigt

Sound und Grafik-Programmierung. Grafische Effekte (z.B. die Verwendung mehrerer Grafik-Bildschirme und Zeichnen in nicht sichtbare Bildschirme

Aufbau und Verwenden der Menü-Leisten in GEM-Accessories in OMIKRON.BASIC

Verwendung der BASIC-internen Multitasking-Befehle

Einige Libraries (Turtle-Grafik, Erweiterungen und Korrekturen zur GEMLib, usw.)

Die Feinheiten des neuen Compilers V 2.0

Umarbeiten von Programmen in GFA-BASIC auf OMIKRON.BASIC

Mit einem Vorwort von Artur Södler, dem Schöpfer von OMIKRON.BASIC

Natürlich befinden sich alle Programme und Beispiele auf Diskette!

HIERMIT BESTELLE ICH___EXEMPLARE
VON "OMIKRON.BASIC".
MIT DISKETTE FÜR DM 49,-
ANRUF GENÜGT: 06196/481811.
MO-FR 9-13 UND 14-17 UHR
SCHRIFTLICHE BESTELLUNG NUR GEGEN

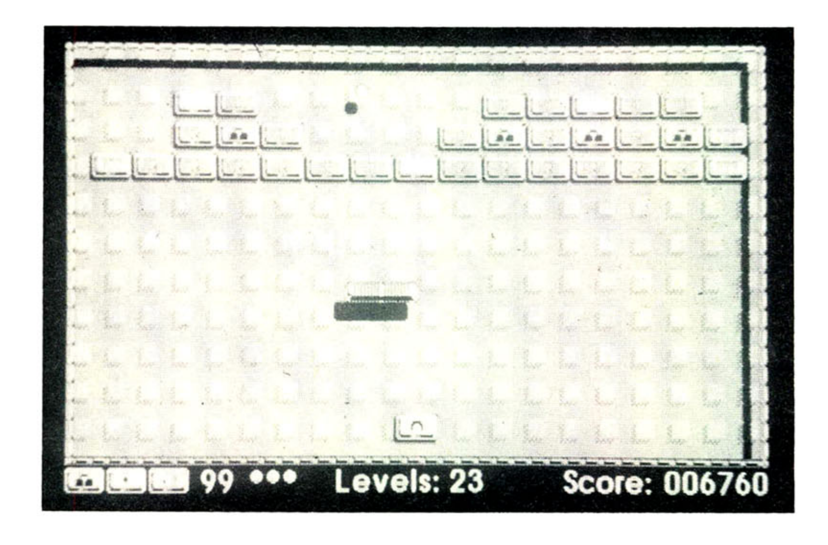
VORKASSE ODER NACHNAHME (VERSANDKOSTEN
DM 5,50: BEI NACHNAHME ZUZÜGLICH DM 4,70)
NAME:____________________
VORNAME:____________________
STRASSE:____________________
ORT:____________________
UNTERSCHR. :____________________



MERLIN COMPUTER GMBH
INDUSTRIESTRAßE 26
6236 ESCHBORN
TEL. 06196/481811

BOLO

BOLO - mit diesem Titel kann man eigentlich wenig anfangen, um so mehr ist jedoch die Grundidee des Spiel jedem Freund dieses Genres bekannt. Da also wieder einmal das ewig junge Breakout bzw. dessen bekannter Nachfolger 'Arkanoid' Pate standen, könnte man nun annehmen, daß sich sonst nichts Besonderes bei BOLO abspielt. Doch das ist weit gefehlt, denn mit diesem Spiel kommen noch einmal neue Varianten hinzu, die den Spielablauf völlig verändern.



Schon gleich beim Einschalten erweckt BOLO ein neues Spielgefühl, denn der Schläger läßt sich frei über den ganzen Bildschirm bewegen. Nur die Spielsteine bilden ein Hindernis und bieten somit auch ein Mittel für verzwickte Situationen: Was passiert, wenn sich der Schläger über der Kugel befindet, selbst umgeben von Steinen, die jedoch ausnahmsweise vom Schläger selbst entfernt werden können, wobei Eile geboten ist, denn die Kugel hopsst inzwischen auf einer Steinbarriere herum und ist nahe daran, durchzubrechen. Solche und ähnliche Situationen machen den Reiz und die Spannung dieses Spiels aus. Dazu kommen noch die Spezialsteine, die im einfachsten Fall mehrmals getroffen werden müssen, bevor sie verschwinden. Erheblich interessanter sind jedoch die Schwerkraft-Steine, nach deren Berührung sich die Kugel nur noch durch kräftige Schläge in der Luft halten läßt, oder die Magnet-Steine, die die Kugel anziehen und ihre Flugbahn stark beeinträchtigen. Andere Steine setzen heimtückische Flugobjekte frei, deren Kontakt man tunlichst vermeiden sollte. Gleiches gilt auch für Steine mit Totenkopfsymbol.

Für Abwechslung und aufregende Situationen ist somit gesorgt, das Spiel ist mit seinen 50 Spielszenen auch nicht einfach zu vollenden. Beim Schlagen der Kugel sollte man nicht immer viel Schwung einsetzen, sondern vielmehr die Tatsache ausnutzen, daß man deren Geschwindigkeit sehr genau steuern kann. Dies geht sogar so weit, daß man sie völlig zum Stillstand bringen kann. Es ist dann in vielen Situationen leichter, Hindernisse abzuräumen, und man läuft nicht Gefahr, die andernfalls hektisch herumfliegende Kugel zu verfehlen.

BOLO ist ein zeitloses und deshalb immer wieder spannendes Spiel, dessen Herausforderung man sich gerne stellt. Die Szenen haben, entgegen den Vorbildern, keine feste Reihenfolge, so daß auch keine Ermüdungserscheinungen aufkommen, wenn das Spiel öfter gespielt wird. Das Spiel fordert neben einem guten Reaktionsvermögen auch gute "Abräum"-Strategien, wobei man sich durch Abbremsen der Spielgeschwindigkeit Luft zum Nachdenken verschaffen kann.

Application Systems Heidelberg, die bisher durch professionelle Programme wie Megamax C und Signum von sich reden machten, haben mit BOLO ihr erstes Spiel veröffentlicht. Es wird jedoch nicht das letzte sein, denn es ist erst der Anfang der kommenden Spiele-Serie. Man kann also gespannt sein, was als nächstes kommen wird.

(Markus Nerding)



JINXTER



Slang bis zum Absturz

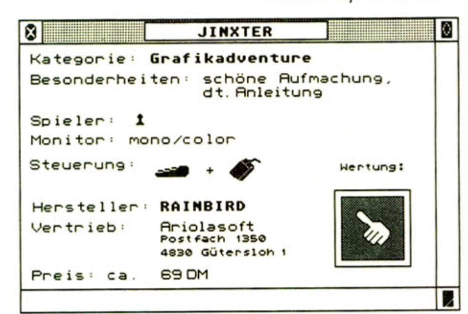
Ich habe einmal einen Computer und ein Spiel, das sich "Guild of thieves" nannte, in die Finger bekommen und mir diese auch in zwei Nächte harter Arbeit mit meinem Freund Max - I don`t understand asshole in that context - zusammen wundgespielt.

Wir waren wirklich begeistert, starben tausend Tode und freuten uns wie die Schneekönige, wenn wir dem Programm wieder einmal so schöne Formulierungen wie "demented pyromaniac" entlockt hatten. Wir wollten uns niemals wieder trennen von diesem herrlichen Zeitvertreib. Es kam aber natürlich alles ganz anders! Das Leben hatte uns bald wieder und ich wurde mit Rainbird Software bzw. Magnetic Scrolls erst wieder konfrontiert, als man mir in der ST-Redaktion "JINXTER" als direkten Nachfolger der Gilde derDiebe wärmstens ans Herz legte. Ich habe in diesem Moment leider vergessen, daß ich ROCKY XXV ziemlich blöde fand, und daß ich auch den Weißen Hai nur bis Teil III ausgehalten habe. JINXTER ist doch stark der Gilde der Diebe nachempfunden.

Die Geschichte der Geschichte

In Aquitania (Name erfunden) hatte der Zauberer Turani (Name erfunden) einst ein besonderes Armband geschaffen. Es sollte die Bewohner durch seine magischen Kräfte vor den allgegenwärtigen bösen Hexen schützen. Dieser schönen Zeit, in der Glück und Frieden in Aquitania anzutreffen waren, wurde aber von der Hexe Jannedor (Name erfunden) und ihren Helfern ein jähes Ende bereitet, indem diese das Armband auseinandernahmen und die Einzelteile im ganzen Land versteckten. Von diesem Tage an wurde Aquitania wieder von Pech und Unglück heimgesucht. Der Spieler hat nun die ehrenvolle Aufgabe, diese Einzelteile wiederzufinden, um so dem Hexenspuk ein Ende zu bereiten. Keine leichte Aufgabe, denn JINXTER verlangt den Dialog mit dem Computer und setzt ziemlich gute Englischkenntnisse (übelster Slang) voraus. Da hilft auch die deutsche Übersetzung des "Independent Guardian", die der Spielpackung beiliegt, nicht darüber hinweg. JINXTER scheint mir schwieriger zu sein als die Gilde der Diebe, und ich gehe davon aus, daß das Bett wiedereinmal nächtelang kalt bleibt. Ran an den Käse.

Rainer Spirandelli



SUPERSPRINT

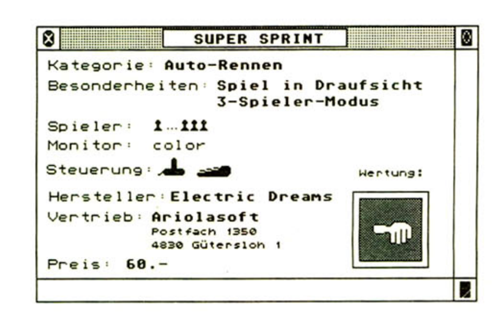
Super Sprint ist ein Autorennen, welches man in 'Draufsicht' spielt, d.h. die ganze Rennstrecke ist auf dem Bildschirm zu sehen und zwar von oben. Eine Besonderheit dieses Spieles ist: Man kann es alleine, zu zweit und sogar zu dritt spielen. Zwei können mit den Joysticks spielen, der dritte im Bunde muß dann die lästige Tastatur benutzen. Zu zweit oder zu dritt kann das Spiel aber besonders Spaß machen, denn dabei kann man seine Mitstreiter von der Bahn drängen um sich dadurch Vorteile verschaffen. Aber nicht nur auf seine Mitstreiter sollte man achten, so gibt es zum Beispiel Ölflecken auf der Straße, die das Auto ins Schleudern bringen, oder Wasserpfützen, die das Auto verlangsamen, oder sogar Windböen, wo das Auto außer Kontrolle gerät. Erfreulicherweise gibt es auch Bonusflächen, die Extrapunkte bringen, wenn man sie überfährt, oder goldene Schraubenschlüssel, die später gegen Sonderzubehör eingetauscht werden können, um die Leistung des Autos zu steigern.

Bremsen gibt es keine! Fährt man zum Beispiel mit mäßiger Geschwindigkeit gegen eine Schranke, so verursacht dies lediglich eine Verlangsamung, bei hoher Geschwindigkeit dagegen kommt es zur Explosion. Man erhält dann von einem Hubschrauber einen neuen Wagen.

Gewöhnungsbedürftig ist allerdings die Steuerung. Da es sich ja um einen Rundkurs handelt, muß bei jeder Kurve wieder umgedacht werden. Erst, wenn die ungewohnte Steuerung einmal erlernt ist, kann das Spiel richtig beginnen.

Die Grafik ist durchschnittlich. Besonders wenn zu zweit oder zu dritt gespielt wird, kommt Freude auf, aber ein Preis von DM 60,-- für das Spiel ist zu hoch.

(AS)



PUBLIC DOMAIN KLASSIKER

Die Diskette 1 bis 85 sind auch weiterhin erhältlich. Schauen sie dazu bitte in einer der vorherigen Ausgaben nach oder fordern Sie die Liste an (frank. Rückumschlag).

Diskette 86

- CARPET : Berechnungsprogramm von 3D-Funktionen. Hidden-Line Algorithmus und Beleuchtungseffekten. Funktionseingabe und frei einstellbare Parameter. (s/w)
- 3D-PLOT : Ähnliches Programm in compiliertem GFA-Basic, mit einigen Extra-Funktionen. (s/w)
- FUNKTIONSPLOT : Funktionseingabe im Programm, verschiebbare Achsen, mehrere Darstellungsarten (s/w)

Diskette 87

- STERNBILD : Zeigt den Sternenhimmel, oder bestimmte Ausschnitte zu verschiedenen Zeiten. Planetennamen auf Anklicken. (s/w und Farbe). Räumliches Rot-Grünbild im Farbbetrieb.

Diskette 88

Druckertreiber
- NEC-EMU : Hardcopytreiber für NEC P5/P6/P7. Erzeugt Hardcopies in 12 verschiedenen Größen und versch. Kontrasten. Enthält Druckerspooler.
- LQ 800 : Hardcopytreiber für EPSON 24 Nadeldrucker in drei verschiedene Größen.
- NEC_CODE : Ein Programm zur Einstellung des NEC P6 /P7 u.ä..
- SIMPLE : Komfortables Hardcopy-Programm mit vielen Extras.
- STX-80 : Hardcopytreiber für diesen preiswerten Thermodrucker.
- WORDPLUSTREIBER : Druckertreibersammlung für folgende Drucker: NL10, FX1000, SP1000I.
- 1ST PROP-TREIBER: 1st Proportional Treiber für folgende Drucker: HR15, FX85.

Diskette 89

- FÜHRERSCHEIN : 500 Fragen zur theoretischen Führerschein-Prüfung.
- BUNDESLIGA : Fußballmanager. Erechnen von Tabellen, editieren aller Spiele von der 1., 2. Bundesliga, bis hin zur untersten Bezirksliga.

Diskette 90

Utilities
- TOPS : The Other Pascal Shell (für PascalPlus), TEMPUSfähig, erzeugt Crossreferenz. Kopieren, Löschen und Umbenennen von Files.
- SIGNUMSHELL : Eine Shell zu Signum. Umschaltung zwischen Signum Editor, den verschiedenen Druckprogrammen, den Zeicheneditoren per Menüleiste. Automatisches Install.
- TEMPELMON 1.8 : DER weitverbreitete Monitor.
- MAXIDISK 4 : Die komprimierende Maxidisk mit max. 4MByte Kapazität. Läuft auf ALLEN TOS-Versionen. Komprimiert automatisch ihren Inhalt. RESETFEST.
- RAMDISK : Gleiches Programm wie MAXIDISK, nur ohne Komprimieralgorithmus.
- COPY : Beide Ramdisk enthalten ein Autocopyprogramm zum kopieren bestimmter Files in Ramdisk.
- RCS.WANDEL : Wandelt *.h Files nach Basic und Assembler.

Diskette 91

Lernprogramme- PAUK : Lerndatei mit komfortabler Steuerung, eigener Dateneingabe, wiederholung falscher Fragen nach Lernkonzept. (s/w)
- HYPERVOC : Sehr schön gestaltetes Vokabellernprogramm. (s/w)
- ST-DIC : Fremdsprachenlernprogramm mit gr. Lateinbibliothek. (s/w)
- QUIZ : Eine Art Trivial Persuite. Enthält Fragen mit drei möglichen aber nur einer richtigen Antwort. Eigene Fragen einbinden möglich. Enthält (ATARI ST- Einsteigerkurs). (s/w)

Diskette 92

Utilitydisk
- SUPERBATCH : Batch-Prozessor: bedingte Batchbearbeitung (if) / Batch Protokoll auf Bildschirm, Drucker, Datei / Verzweigung, Fehlerbehandlung.
- ST-KLICK : Multifunktionsaccessoir: Notizen, Wecker, Drucker Init, Schreibmaschine, Kalender, Taschenrechner (s/w)
- XREF : Crossreferenz zu C
- TERM : Terminalprogramm mit Xmodem, Vollduplex, u.v.a.m.
- ALFAFORMAT : flexibles Formatierprogramm (9-11 Sektoren) (bis 86 Tracks).

Diskette 93

- ADR_2 : Semiprofessionelles Adressverwaltungsprogramm mit Serienbrieferstellung (mit und ohne 1st Word). Editierbare Druckeranpassung (Steuerzeichen, Zeichenwandlung). erzeugt Adress- Telefon und Geburtstagslisten. Druckt Labels, Etiketten nach freidefinierbarem. Format. (s/w)
- DAT_TEXT : Adressverwaltung für Mehrpersonenzugriff auf eine Datei. Datenzugriff nur durch Passwort (s/w)

Diskette 94

Spieledisk
- MACPAN : Eine schnelle Pacman-Version für den ST. Per Joystick ins Labyrinth auf der Suche nach Powerpills. Mit Empfehlung der Redaktion.
- SOLITÄR : Dreidimensionales Solitär mit sehr ansprechender Grafik.
- GALAXY + SPRENGMEISTER : Brettspiele zur Förderung der Gehirnaktivität (s/w)

Diskette 95

Spieledisk
- DIAMOND MINER : Ein Spiel ähnlich Boulder Dash. Sehr schnelle Grafik und 30 Level. Eingebauter Feldeditor (s/w)
- SNAKE : Zwei Schlangen versuchen sich den Weg abzuschneiden.(s/w)
- MINENFELD : Ein wagemutiger Schatzsuchen begibt sich durch verschiedene Minenfelder und muß versuchen die Minen zu umgehen.(s/w)
- MISSLE : Missleattack (s/w)

Diskette 96

Spielesammlung
- ANDURLIN : Geschicklichkeitsspiel mit 75 Bildern. (s/w)
- GILGALAD : Arcade Adventure mit 224 Räumen. (s/w)
- INVADORS : Der Spieleklassiker. Außerirdische Raumschiffe greifen die Erde an. (s/w)
- MINIGOLF : Minigolf auf dem ST. 18 Bahnen, Steuerung (Richtung, Anschlagstärke) per Maus. (s/w)

Diskette 97

- TRASH : The First Trash-Groove Adventure. Textadventure mit deutschen Befehlen. Parodie auf die Musik- und Punkszene.
- LABYRINTH : Im Labyrinth auf der Suche nach dem Ausgang. Sehr schöne 2D/3D-Darstellung. Schnell und ruckfrei. (s/w)

Diskette 98

VAX GAMES Spielesammlung
- SNAKE : Eine Schlange auf Nahrungssuche.
- QIX : Sie müssen 75% des Bildschirms einzäunen ohne gebissen zu werden. (Ähnlich STIX)
- NIBBLER : Eine Schlange schlängelt sich durch ein Labyrinth.
- PACMAN : Kommentar überflüssig
- DOORS : Pacman Variante mit Hindernissen
- ROBOT : Vosicht Roboter

Diskette 99

- CNC SIMULATION : Simulation einer CNC gesteuerten Fräsmaschine. Programmierbar. Grafische Anz. (s/w)

Diskette 100

- ST-SPEECH : Sprachausgabe auf Softwarebasis. Wandelt englischsprachige Texte automatisch in verständliche Lautschrift. (s/w)
- FORTRAN SHELL : Eine Shell für Fortran 77. Bequemes Aufrufen von Compiler, Linker, und anderen Programmen.
- LIFE IS LIFE : Life-Simultionsprogramm. Sechs verschiedene Feldgrößen, Drehen, Verschieben, Kopieren von Blöcken wie im Malprogramm, Festlegen der Fortpflanzungsregeln, hohe Geschwindigkeit. (s/w)
- FASTLIFE : Life-Simulation in Volksforth. Wahnsinnige Geschwindigkeit. (s/w)
- GEMFRAC : Berechnung fraktaler Landschaften in Farbe und s/w.
- SHARP : Basic Programme für Sharp-Basicrechner. Geeignet für Direktüberspielung mit Interface.

Diskette 101

Wissenschaftliche Anwendungen (Schwerpunkt Chemie Medizin)
- LAOKOON : Sumulation von NMR (Magnetische-Kern-Resonanz-Spektoskopie). Bestimmung von chemischen Verbindungen u. Kopplungsarten. (s/w)
- IR : Infrarot-Spektroskopie. Programm zur Strukturanalyse. (s/w)
- PLOTTER 1.7 : Grafische Darstellung von Werten. Lagrange- Interpolation, Spline oder Regressionsgrade. (s/w)
- PLOTTIT : Kurvenplotter. Direkte Funktionseingabe mit sofortiger Syntaxprüfung. Nullstellenberechnung. (s/w)
- GANGLION : Programm zur Simulation eines neuralen Netzes (s/w)

Diskette 102

Programmierutilities
- FILESELECT : Eigene flexiblere Fileselectroutine. Zum Einbinden in eigene Programme.
- PRINTF : Verbesserte PRINTF-Routine für Megamax C.
- RCS_ICN : Generiert ein ICON aus einem Bildschirmbereich

Diskette 103

Adimens Utilities Für ADIMENS und ADITALK
- ONE_TO_ADI : Konvertierung von DB_Master_One Dateien zu ADIMENS. Mit Zeichenanpassung.-
MAT_TO_ADI : Konvertiert Datamat-Dateien ins ADIMENS-Format.
- KONV_ALT_NEU : Programm zum Anpassen von Datensätzen der Version 1.6 zu Version 2.1.

Diskette 104

- ST_CALC :Kalkulationsprogramm mit vielen Features. 2600 Zellen. Variabler Bildaufbau. Formeleingabe. FolgendeOperationen: (), *, +, -, ^, /, SUM, AVE, STA, MIN, MAX, ABS, INT, RND, LOG, EXP, CLOG, SQR
- LOHNSTEUER : Berechnet die Lohnsteuer für die Jahre 1977 und 1988
- ANLAGE : Verwaltungsprogramm von Anleihen, Aktien, Investmentanteile, u.a. (s/w)
- DEPOT : Aktiendepot. Nützlichesprogramm für Finanzgenies.
- AKTIE : Auch dieses Programm dient der Aktienverwaltung.

Diskette 105

Technische Anwendungen
- REGELPRO : Simulation einer Regelstrecke. Grafische Darst.. (s/w)
- SCHALTPLAN : Entwickeln von elektronischen /elektrischen Schaltplänen. Viele Features und Schaltsymbole. (s/w)
- ROTAX 2.0 : Rotations- und Animationsprogramm IN 3D. (s/w)

Diskette 106

Musikprogramme
- MUSIK : Komfortables Musikprogramm. Noteneingabe mit Maus. Dreistimmig. Erzeugt Soundstring, für Einbau in eigene Programme (XBIOS(32)). Viele Demostücke. (s/w)
- SOUNDCOMPUTER : Soundkreation mit Soundchip. Lechte Eingabe. Übernahme in Basic. (s/w)

Diskette 107

- NAPOLEON : Strategiespiel ähnlich'Risiko'. (s/w)

Diskette 108

Druckerutilities (P6 u.a.)
- SIG PIC : Grafikeinbindung in das alte SIGNUM! V1.0. Versch. Bild-Größen.
- ZS Editor 24 : Zeichensatzeditor für 24-Nadeldrucker. Down-Load. Für alle Programme (z.B. Wordplus). (s/w)
- MANAGER : Accesoir zum Einstellen eines P5/6/7-Druckers.
- PRINT_OUT : Ausdruck von Source-Code (Pascal oder C) mit Zeichenanpassung und beliebige Schriftartwechsel.
- POSTER : Druckt Bild in vierfacher Größe. (s/w)

Diskette 109

Wordplus-Utilities u.a.
- INHALT : Erzeugt Index-Liste von 1stWord-Texten. - INDEX_PLUS : Erzeugt Liste von Seitenindex, Zeilenlinealen und allen Bildern.
- BT_Konvert : Konvertiert 1st_Word-Texte nach Beckertext.
- TYPEWRITER : Schreibmaschinenkurs in über 40 Lektionen.(s/w)
- MORSE : Morsetrainingsprogramm.

Diskette 110

Utilities
- R_COPY : Automatischen Backup von Ramdisk auf Diskette. Incl. Source in Pascal.
- PASHELP : Crossfeferenzprogramm für Pascal.
- UNIDRUCK : Druckprogramm mit Sonderzeichenanp., mehrere Drucker. Zeilenummerierung. Incl. Source in Pascal. - FONT : Programm zum Einbinden von DEGAS-Fonts in Pascal-Programme. Incl. Source in Pascal.

Diskette 111

Utilities
- BASICREF : Crossreferenz für GFA-Basic. Incl. Source in GFA-Basic. (s/w)
- MSK_Edit: Maskeneditor für GFA-Basic. Generiert Quelltext (Form Input). Incl. Source in GFA-Basic. (s/w)
- RETTEN: Utilitie für lange Dateien. Incl. Source in Fortran.
- SECO:Bildausschnittsbibliotheksverwaltung (für PUT GET von GFA). Incl. Source in GFA-Basic.

Diskette 112

Erdkunde
- WORLD: Lernprogramm. Fragt alle Länder und Hauptstätte der ganzen Weltab. Landkarten von BRD, USA, Mittelamerika, Südamerika, Europa, Asien, Afrika und Ozeanien. (s/w)

Diskette 113

- ZEITMANAGER : SemiprofessionelleTerminplanverwaltung. Eigener Desktop, Terminerinnerung, Listendruck, Timingverwaltung, fixe Termine, Kalender. (s/w)

Diskette 114

Spiele
- METROPOL : Wirtschaftsspiel. RettenSie die Wirtschaft ihres Landes. 'Auf der Suche nach der Wende'.(s/w)
- DALLAS : Öl, Öl, Öl, Geld, Geld, Geld, JR, JR, JR (f)

Diskette 115

Spiele
- SHERLOCK: Klären Sie als Holmes den Mord.
- EL BOZO CITY: Textadventure mit vielen Gags.
- SCRIBBLE : Schöne Scrabble-Version für den ST. (s/w)
- KREUZWORT : Generierung eines Kreuzworträtsels. (s/w)

Diskette 116

- µEMACS 3.8 : Deutsche Version. Eigene Macrosprache, Textverschlüsselung, Mailmerge, mehrere Screens, u.v.a.m.

Diskette 117

Terminalprogramme
- UNITERM : Exclusives Terminalprogramm. VT200, VT102, VT100, VT52,4010, u.a. Softscroll, Grafikübertragung, Funktionstastenbelegung,..

Diskette 118

Utilities
- M_COPY : Universelles Kopierprogramm bzw. Diskmonitor.- LOADER : Aktiviert oder deaktiviert Accesoiries bzw. Autostartprogramme.
- VIEW : Programm zum Anzeigen von ASCII-Files. Vor- und Rückblättern per Tastendruck.
- SUPERSHELL : Universelle Shell. Aufruf beliebiger Programme aus der Menüleiste. Source in GFA-Basic.
- LATTICE : diverse Utilities: HEXDUMP, Ausdruck mit Zeilennummern, Symboltabellen, Archivierung und Pflege von Modulbibliotheken.

Diskette 119

Etiketten Drucker
- TAPE LABEL: Druckt Cassettenlabels.
- LABEL : Komfortabler Disketikettendrucker. Übernahme eines Bildes per Mausklick. Eigene Bilder integrierbar. (s/w)
- PASTE ADRESS : Adressaufkleber in Miniformat (1.5cm*2.5cm)

Disketten 120

- WISSEN SIE ES? : Quizspiel mit vielen Fragen aus verschiedenen Gebieten. Ähnlich 'Trivial Pursuit'. Viele Fragen aus verschiedenen wissensgebieten, eigene integrierbar. (s/w)

Diskette 121

Spiele
- GO_UP : Loderunner-ähnliches Spiel. Auf vielen Mauern, Leitern und Seilen müssen Sie den Verfolgern entkommen. Viele Level. Editor zum Entwickeln eigener Levels enthalten. (s/w)
- MAZE_EDIT : Editor zu dem Spiel 'MIDI MAZE'.
- F1 MANAGER : Verwaltung eines Formel-Eins-Rennstall. (s/w)
- OELIMPERIUM : Managen einer Ölfirma. (s/w)

Diskette 122

Spiel
- PD_BOLO : Sie kennen Arkanoid, dann sollten Sie auch PD_Bolo kennen. Interessante Break-Out Variante.

(s/w) - Nur Monochrom
(f) - Nur Farbe
kein Kürzel - Farbe und Monochrom

Sonderdisks

Die folgenden Programme sind nicht Public-Domain.
Sie können aber bei uns bezogen werden.

A.) TOS : Die letzte Disketten-Version vom 6.2.1986. Ältere Versionen laufen nicht problemlos.
Unkostenbeitrag samt Diskette DM 15.-
B.) RCS : Das Resource-Construction Set aus dem ATARI-Entwicklungspakets. Unverzichtbar bei der GEM-Programmierung.
Unkostenbeitrag samt Diskette DM 15.-

Bis auf die Höhe des Unkostenbeitrages gelten die gleichen Versandbedingungen wie bei der Public-Domain Software.

VERSANDBEDINGUNGEN

Sämtliche Disketten können ab dem Erstverkaufstag der ST-Computer direkt bei der Redaktion bezogen werden. Wir haben für Sie den schnellstmöglichen Versandservice eingerichtet. Lieferung innerhalb einer Woche.

1. Schriftliche Bestellung

- Der Unkostenbeitrag für eine Diskette beträgt DM 10.-
- Bezahlung nur per Scheck oder Nachnahme (Im Ausland nur Vorrauskasse möglich)
- zuzüglich DM 5,00 Versandkosten (Ausland DM 10.-)
- bei Nachnahme zuzüglich DM 3,70 Nachnahmegebühr
- Legen Sie bitte, falls zur Hand, einen Aufkleber mit Ihrer Adresse bei.
- Bitte fügen Sie keine anderen Bestellungen oder Anfragen bei.

2. Anruf genügt

'MERLIN'-Computer GmbH
ST-Computer Redaktion
Tel.: 0 61 96 / 48 18 11
Von Mo-Fr 9.00 bis 17.00 Uhr

Die Bezugsadresse lautet:
'MERLIN'-Computer GmbH
ST-Computer Redaktion
'PD-Service'
Postfach 5969
D-6236 Eschborn

Bei Fragen bezüglich der Programme stehen wir Ihnen gerne zur Verfügung.

PUBLIC DOMAIN SERVICE

Liebe Leser,

jeden Monat erhalten wir viele neue PD-Programme von Lesern, aus denen wir die neuen Disketten zusammenstellen. Auch diesen Monat können wir Ihnen wieder einige interessante Programme bieten, so z.B. eine neue Fileselektorbox, die sich problemlos ins System einklingt und somit in allen Programmen die normale Box ersetzt. Auch für die wissentschaftlichen Anwender ist wieder einiges vorhanden. Last bot not least die Spieldiskette. Diesmal nur eine, aber diese mit einem sehr schönen Spiel einiges an Geschick erfordert. Doch schauen Sie besser selbst in der Liste, was Ihnen gefällt.

Ihre ST-Computer Redaktion

DIE NEUHEITEN

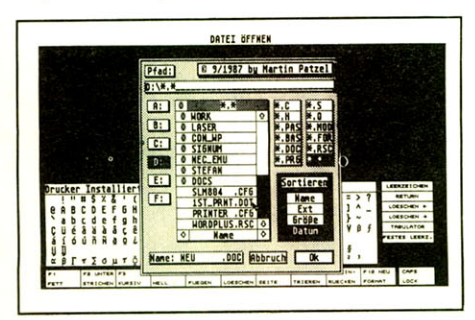


Utilities

- Hyperformat Formatierprogramm von Claus Brod. Holt das letzte aus der Floppy heraus.



- FSELECT Eine überaus geniale neue Fileselectorbox, die sich ins System einklingt und von jedem Programm anstatt der alten benutzt wird. Mit Vorbelegung, mehreren Laufwerken, anpassbar an persönliche Wünsche.

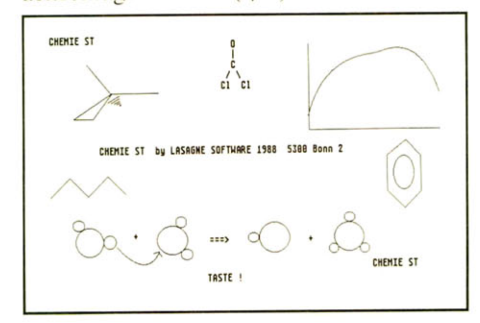


- Diskmon Diskettenmonitor. Einfaches Bedienen, Sektoreditieren, Suchen u.v.a.m. (s/w)
- PC_Ass Assembler für Sharp-Taschencomputer. Das Programm assembliert und disassembliert Sharp-Maschinencode.



Chemie

CHEMLIB Dateiverwaltung für chemische Elemente, Verbindungen, Moleküle, Reaktionen und Gesetze. Erstellung eines kompletten Chemienachschlagewerkes. (s/w)

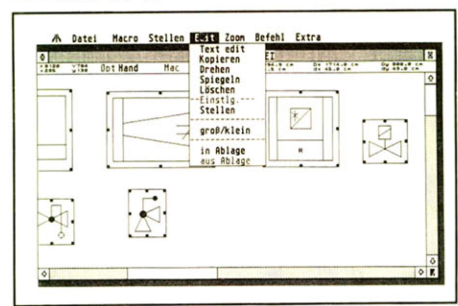




Zeichenprogramm

CAD 2

Objektorientiertes Grafikprogramm. Im Gegensatz zu rein pixelorientierten Zeichenprogrammen, verwaltet CAD 2 alle Elemente als Objekte. Diese können ohne Auflösungsverlust vergrößert, verkleinert, gedehnt oder gestaucht werden. Grafische Grundfunktion, Spiegeln, Rotieren, Texteinbindung, Zoom.... (s/w)





- CUBE HACK Terminalprogramm mit integrierter Programmiersprache. Variablen, Wertezuweisung, Ein-Ausgabe, Kontrollstrukturen, Funktionstastenbelegung, Dateibefehle, Druckersteuerung, Senden, Empfangen, Rechenfunktionen. Ausführen von Batch-Befehls-Dateien. Viele Demos. Anwendungen: z.B. Automatisches Einloggen in Mailboxen.
-OR Spread Spreadsheetähnliches Programm zur Linearoptinmierung, Netzplanentwicklung, Kürzeste Wege Berechnung. (Matritzen bis zu 500*500) (s/w)



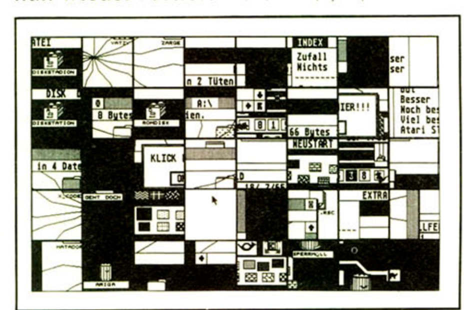
Wissenschaftliche Anwendungen

- Planet Himmelskörperberechnungen. Berechnet die Ephemeriden des Mondes und der Planeten, sowie anderer heliozentrischer Himmelskörper. (Grafische Darstellung des Sternenhimmels, Horizont). Anzeige aller Sterne und Sterndaten, zu bestimmten Zeiten, verschiedenen Orten und Blickrichtungen. Animation derselben.
- Turing Realisierung eines Turing-Maschinen-Modells (s/w)
- Zust Analyse und Simulation linearer Regelkreise im Zustandsraum (s/w)

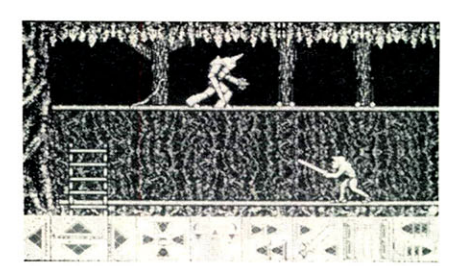


Spielesammlung

- Horrorschloß Sie müssen sich durch verschiedene Gänge bewegen. doch Vorsicht: Geister und Falltüren behindern Sie. Weiterhin enthalten sind Strahlenwände, Rollbänder, Leitern, Rutschstangen, wichtige Schlüssel und Transporter.
- Inversi Das bekannte Spiel, bei dem Sie mehr Steine umdrehen müssen als Ihr Gegner (Rechner oder zweiter Spieler). (s/w)
- Invation Space Invadors mit 3-D Vektorgrafik. Das Spiel für Aktion-Freunde. (s/w)
- Zarge Ein 32000K-Bild wird in viele Teile geteilt, vermischt und muß nun wieder sortiert werden. (s/w)



In der nächsten ST-Computer lesen Sie unter anderem



Bauen Sie Ihren eigenen Multisync!

Als NEC den ersten Multisync-Monitor herausbrachte, war es vor allem für die ATARI Besitzer interessant, da alle drei Auflösungen des ST mit einem Monitor benutzt werden können. Wir zeigen Ihnen (siehe Bild), wie Sie Ihren ATARI-Monitor SM124 mit etwas Geschicklichkeit und Geduld ebenfalls zu diesem Ergebnis bringen können. Natürlich werden keine Farben dargestellt, sondern nur Graustufen. Sicherlich ist es aber trotzdem ein Leckerbissen und eine billige Alternative für alle, die bisher auf die niedrige und die mittlere Auflösung verzichten mußten.

Schönere Programme durch GEM-Zeichensätze nach Wahl

GEM bietet einige sehr interessante Möglichkeiten, die auf dem ST leider noch gar nicht richtig ausgenutzt werden. Ein wichtiger Punkt dabei sind die Textdarstellungen. Da jeder Text im Grafikmodus ausgegeben wird, bietet GEM (VDI) die Möglichkeit, verschiedene Textattribute (fett usw.) einzustellen und die Textgröße zu wählen. Davon machen ja auch schon viele Programme (v.a. im Grafikbereich) reichlich Gebrauch. Die Entwickler von GEM sind aber noch einen sehr großen Schritt weitergegangen. Man kann außer den Systemzeichensätzen noch andere Fonts von Diskette laden (sollte man wenigstens können). Dabei tritt nur ein großes Problem auf: die Implementation auf dem Atari ST.

ST-Kontor im Büro

Ein neues Softwarepaket für den Einsatz im kaufmännischen Bereich brachte der bekannte SYBEX-Verlag jetzt auf den Markt. ST-Kontor besteht aus mehreren Moduln, die je nach Bedarf hinzugekauft werden können. Zur Zeit sind drei dieser Moduln erhältlich, die wir Ihnen gerne als Alternative zu den bereits auf dem Markt befindlichen Programmpaketen vorstellen wollen.

Extended VT 52-Emulator

Daß man den eingebauten VT 52-Emulator des ATARI ST noch deutlich verbessern kann, wollen wir Ihnen anhand einer Mini-Serie zeigen. Serie deswegen, weil wir niemand ca. 1800 Programmzeilen in einer Ausgabe zumuten wollen. Doch das fertige Programm (nur 4,5 K lang) rentiert sich, da man Escape-Sequenzen von A - Z und somit viele neue Features erhält (z.B. 2-6 mal schnellere Bildschirmausgabe, erstmalige Grafikfähigkeit für problemlose Übertragung bei DFÜ und vieles mehr). Natürlich bleibt das Programm (abwärts-) kompatibel zum VT 52-Standard.

Änderungen vorbehalten !

Die ST-Computer Ausgabe 4/88 erscheint am 31.3.1988

Fragen an die Redaktion

Ein Magazin wie die ST-Computer zu erstellen kostet sehr viel Zeit und Mühe. Da wir ja weiterhin vorhaben, die Qualität zu steigern (ja, auch wenn das manchmal danebengeht), haben wir Redakteure ein großes Anliegen an Sie, liebe Leserinnen und Leser:

Bitte haben Sie Verständnis dafür, daß Fragen an die Redaktion in Zukunft nur noch zu bestimmten Zeiten beantwortet werden können. Wir stehen Ihnen zu folgenden Terminen telefonisch zur Verfügung:

Dienstag und Donnerstag von 14 - 17 Uhr

Vielen Dank für Ihr Verständnis

Ihre Redaktion

Impressum

ST Computer

Chefredakteur:
Uwe Bärtels (UB)

Redaktion:
Harald Egel (HE)
Marcelo Merino (MM)
Harald Schneider (HS)

Redaktionelle Mitarbeiter:
Claus Brod (CB) Jürgen Leonhard (JL)
H.P.Labude (HP) Markus Nerding (MN)
Stefan Höhn (SH) Chr. Schormann (CS)
Raymund Hofmann (RH) Andreas Suchy (AS)
Oliver Joppich (OJO) Jörg Wilhelm (JW)

Autoren dieser Ausgabe:
D.Brockhaus Dr.V.Kurz
I.Brümmer J.Landler
W.Cramer D.Rabich
I.Eichenseher R.Tolksdorf
A.Esser C.Vogler
J.Falkenberg Th.Weinstein
U.Hilwerling K.Wilczek
J.Kleveman

Public Relations:
Claus P. Lippert (Leitung)
D.de la Fuente (UK)
L.Hennelly (Nordamerika)

Redaktion: "Merlin" Computer GmbH
Postfach 59 69
Industriestr. 26
6236 Eschborn
Tel.: 0 61 96/48 18 11
FAX : 0 61 96/4 11 37

Verlag:
Heim Fachverlag
Heidelberger Landstr. 194
6100 Darmstadt 13
Tel.: 0 61 51/5 60 57
FAX : 0 61 51/5 56 89 + 5 60 59

Verlagsleitung:
H.J.Heim

Anzeigenverkaufsleitung:
U.Heim

Anzeigenverkauf:
Kyriakulla Magaritis

Anzeigenpreise:
nach Preisliste Nr.3, gültig ab 1.1.88
ISSN 0932-0385

Layout:
Fabian & Mayer

Fotografie:
R.Spirandelli, Archiv

Produktion:
K.H.Hoffmann, K.Schultheis, B.Failer, S.Failer
Druck:
Ferling Druck

Lektorat:
V.Pfeiffer

Bezugsmöglichkeiten:
ATARI-Fachhandel, Zeitschriftenhandel, Kauf- und Warenhäuser oder direkt beim Verlag

ST Computer erscheint 11 x im Jahr
Einzelpreis: DM 7,-, ÖS 56,-, SFr 7,-
Jahresabonnement: DM 70,-
Europ. Ausland: DM 90,-
Lüftpost DM 120,-

Manuskripteinsendungen:
Programmlistings, Bauanleitungen und Manuskripte werden von der Redaktion gerne angenommen. Sie müssen frei von Rechten Dritter sein. Mit seiner Einsendung gibt der Verfasser die Zustimmung zum Abdruck und der Vervielfältigung auf Datenträgern dem Heim Verlag. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte wird keine Haftung übernommen.